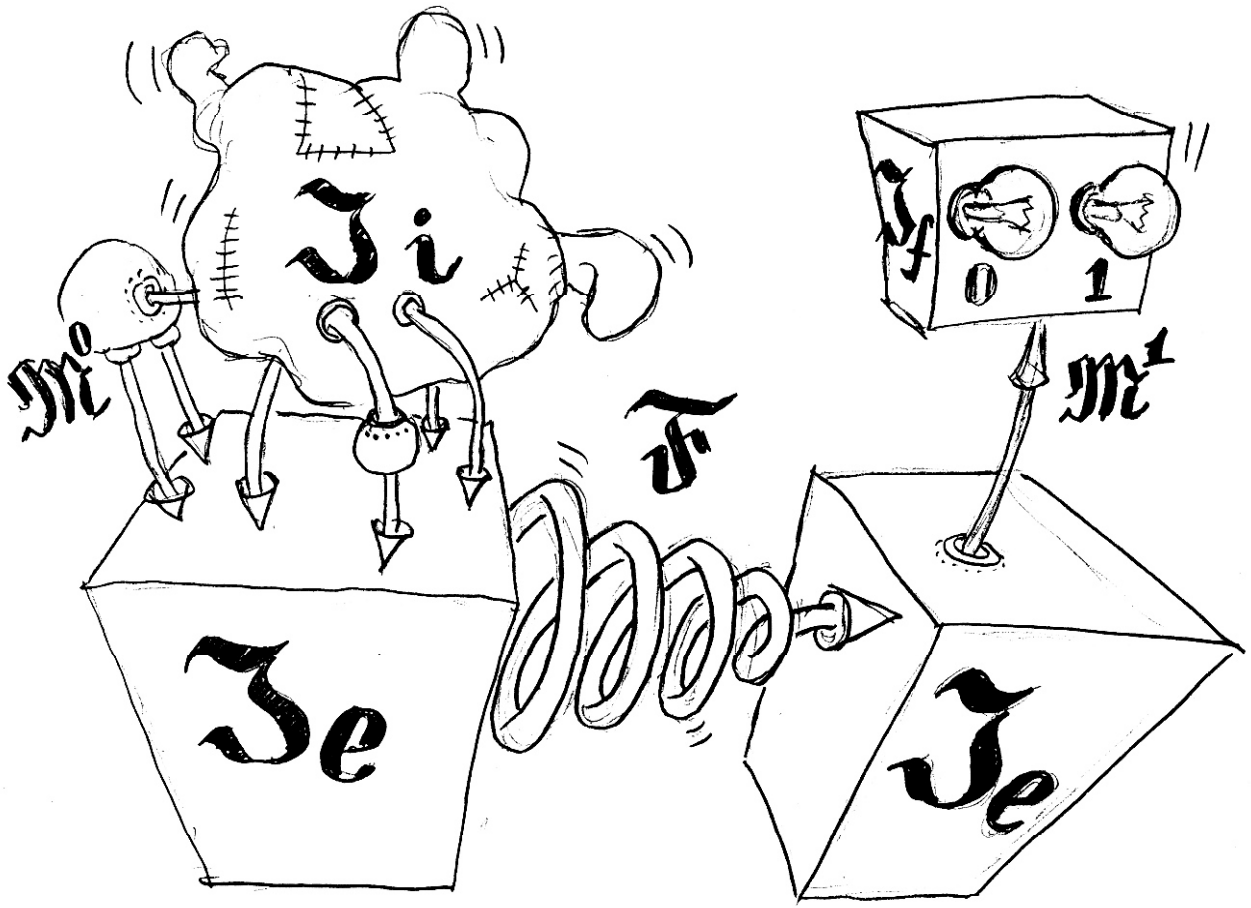
Overfitting and generalization ability of data mining models



Overview

- Generalization ability: definition
- Justification:
  - Data mining could be useful in Search
  - Demo: overfitting always happens
- Mathematical frameworks
  - Statistical learning theory (SLT)
    - Computational Learning Theory? (COLT)
  - Combinatorial learning theory

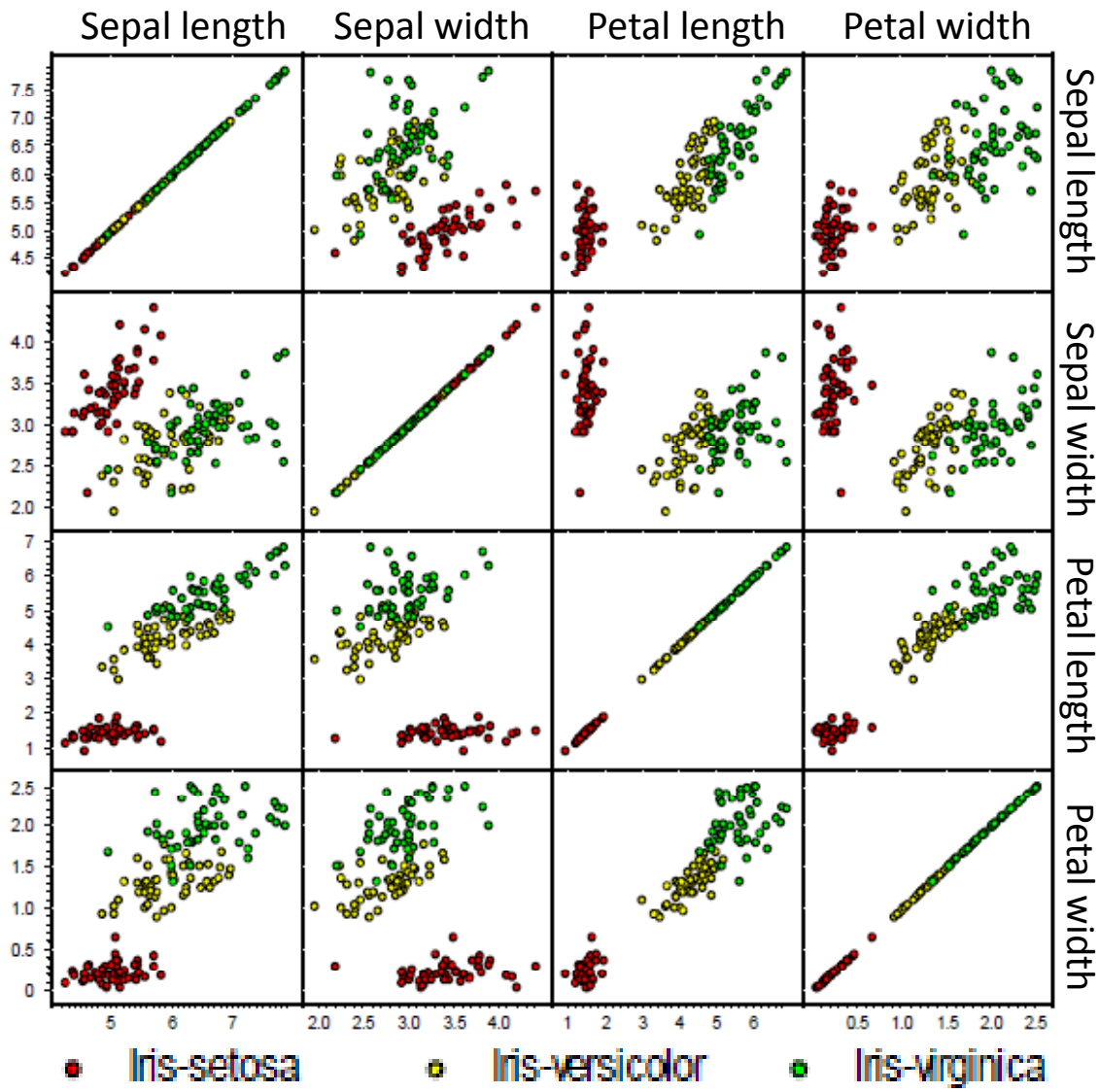
Alexander Frey

- Forecsys company
  - Forecasting and recognition systems
- Search Core Test team in FAST
- Master's thesis: "Exact bounds of overfitting probability for symmetric sets of predictors"

Classification task

Example:

- Iris flowers classification
- 3 values of target class
- 150 items in dataset (sample)
- 4 features
- UCI Machine Learning Repository



Classification task

- Feature matrix
  - Rows = Items
  - Columns = Features
  - Last column = Target feature

	A	B	C	D	E
1	5.1	3.5	1.4	0.2	Iris-setosa
2	4.9	3	1.4	0.2	Iris-setosa
3	4.7	3.2	1.3	0.2	Iris-setosa
4	4.6	3.1	1.5	0.2	Iris-setosa
5	5	3.6	1.4	0.2	Iris-setosa
6	5.4	3.9	1.7	0.4	Iris-setosa
7	4.6	3.4	1.4	0.3	Iris-setosa
8	5	3.4	1.5	0.2	Iris-setosa
9	4.4	2.9	1.4	0.2	Iris-setosa
10	4.9	3.1	1.5	0.1	Iris-setosa
11	5.4	3.7	1.5	0.2	Iris-setosa
12	4.8	3.4	1.6	0.2	Iris-setosa
13	4.8	3	1.4	0.1	Iris-setosa
14	4.3	3	1.1	0.1	Iris-setosa
15	5.8	4	1.2	0.2	Iris-setosa
16	...	...	...	...	...

- Sample $\mathrm{X}^L = \{(x_i, y_i)\}_{i=1}^{L}$
- Predictor $f(x_i)$ should guess y_i
- Learning algorithm $\mu(\mathrm{X}^L)$ should tune good f

Classification task

Term	Notation	Synonyms
Items	x_i	Objects, Instances
Features	$\overrightarrow{x_i}$	Properties, Attributes
Class	y_i	Value of Target feature
Predictor	f $f(x_i) \approx y_i$	Hypothesis, Function, Model, Algorithm, Classifier
Learning algorithm	μ $\mu(\mathrm{X}^\ell) = f$	Learning method
Sample	$\mathrm{X}^L = \{(x_i, y_i)\}_{i=1}^L$	Dataset

Classification Loss Function

- But when predictor f is good?
- Loss function should be small:

$$\mathcal{L}(\mathrm{X}^L, f) = \sum_{i=1}^{L} [f(x_i) \neq y_i] \to \min_f$$

- But when learning algorithm μ is good?

Classification Loss Function

- But when predictor f is good?
- Loss function should be small:

$$\mathcal{L}(\mathrm{X}^L, f) = \sum_{i=1}^{L} W(x_i)[f(x_i) \neq y_i] \to \min_f$$

- But when learning algorithm μ is good?

Classification Loss Function

- But when predictor f is good?
- Loss function should be small:

$$\mathcal{L}(\mathrm{X}^L, f) = \sum_{i=1}^{L} W(x_i) W_{f(x_i)}^{y_i} \to \min_f$$

- But when learning algorithm μ is good?

Classification Loss Function

- But when predictor f is good?
- Loss function should be small:

$$\mathcal{L}(\mathrm{X}^L, f) = \sum_{i=1}^{L} W(x_i) W_{f(x_i)}^{y_i} \to \min_f$$

- But when learning algorithm μ is good?
  - Average loss on test sample should be small:

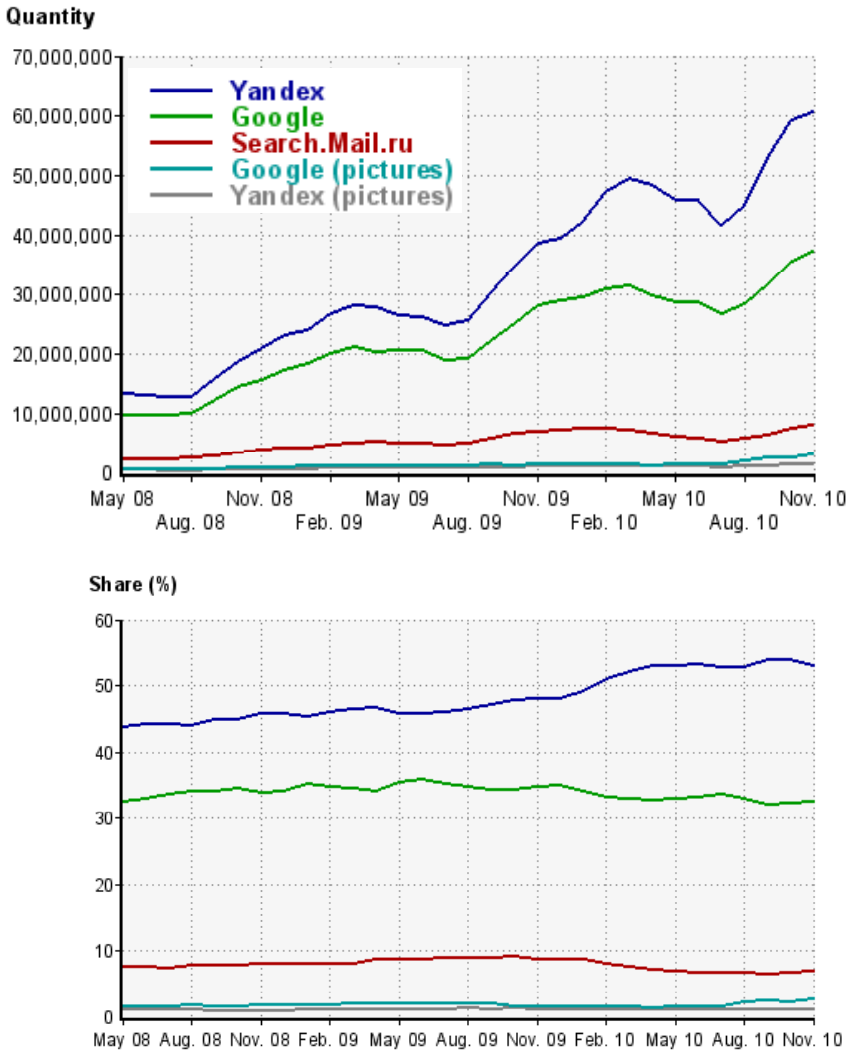
$$CV(X^L, \mu) = \frac{1}{\binom{L}{\ell}} \sum_{X^L = X^\ell \sqcup X^k} \mathcal{L}\left(\mathrm{X}^k, \mu(X^\ell)\right) \to \min_\mu$$

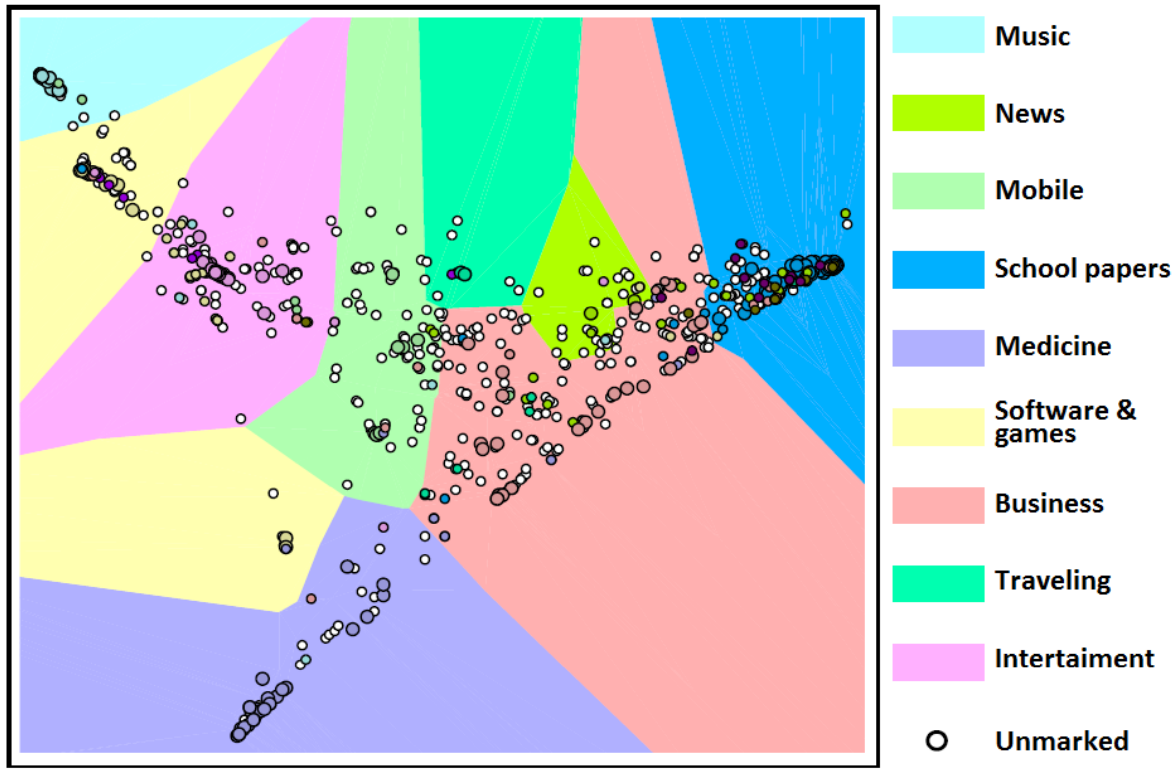
Generalization ability

- is an ability of learning algorithm to tune predictors, that not only performs well on training dataset, but that are also capable of making valuable predictions on test dataset that was completely unavailable during training stage.

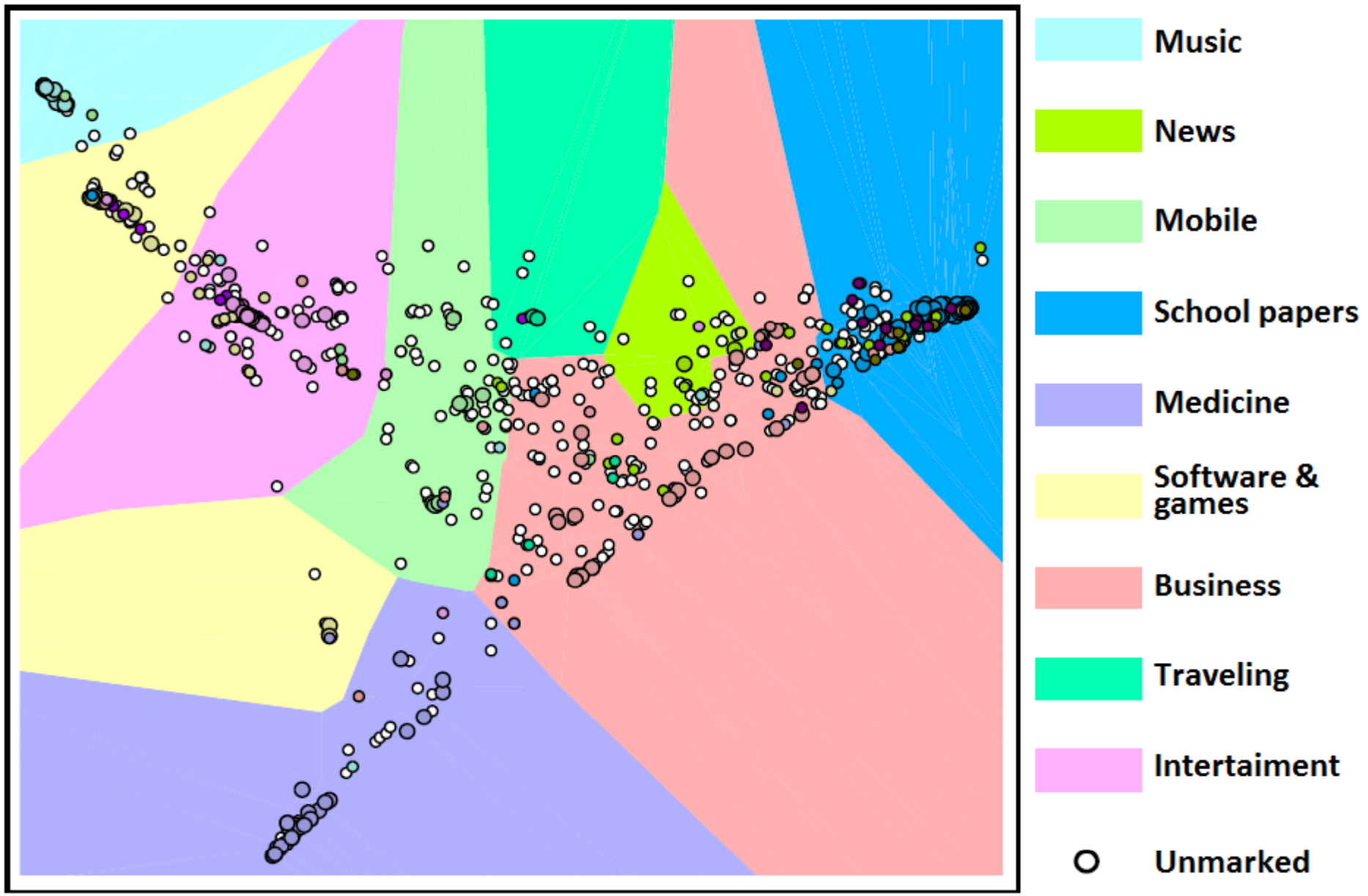
Yandex.ru vs Google.ru

- Most popular Russian Web-Search Provider
  - http://www.liveinternet.ru/stat/ru/searches.html?period=month
- Yandex school of data processing
  - L2R Competition
    - http://imat2009.yandex.ru/en
  - Data Mining courses from top Russian scientists





- Similarity map of internet websites.
- Dataset provided by Yandex company was about visiting 129 600 websites by 14606 users.
- Only 1000 most popular websites are displayed on the map.
- Among them 400 websites were manually categorized to one of nine classes.
  - But this information was unavailable for algorithm that calculates point coordinates on the map.
  - It was used only to point items and background into different colors.


Music
News
Mobile
School papers
Medicine
Software & games
Business
Traveling
Intertaiment
Unmarked

Clustering Task

- Classification task = supervised learning:
  - Sample with target classes $\mathrm{X}^L = \{(x_i, y_i)\}_{i=1}^L$
- Clustering = unsupervised learning:
  - Sample without target classes $\mathrm{X}^L = \{x_i\}_{i=1}^L$
- Predictor $f(x_i)$ still need to assign target class
- We want:
  - Small distances within cluster
  - Big distances between clusters

Clustering Loss Function

- Distance between items $\rho(x_i, x_j)$
  - Could also be calculated based on Feature Matrix
- Distance between set of items $R(\{x_i\}, \{x_j\})$

Closest neighbor distance	$R\ (W,S) = \min_{w \in W, s \in S} \rho(w,s);$
Most distance neighbor distance	$R\ (W,S) = \max_{w \in W, s \in S} \rho(w,s);$
Average distance	$R\ (W,S) = \frac{1}{\lvert W \rvert \lvert S \rvert} \sum_{w \in W} \sum_{s \in S} \rho(w,s);$
Distance between centers	$R\ (W,S) = \rho^2\left(\sum_{w \in W} \frac{w}{\lvert W \rvert}, \sum_{s \in S} \frac{s}{\lvert S \rvert} \right);$
Ward distance	$R\ (W,S) = \frac{\lvert S \rvert \lvert W \rvert}{\lvert S \rvert + \lvert W \rvert} \rho^2\left(\sum_{w \in W} \frac{w}{\lvert W \rvert}, \sum_{s \in S} \frac{s}{\lvert S \rvert} \right);$

Mutual clustering

- Online movie shop problem:
  - User bought “Gone with the wind”. Is it reasonable to propose him to buy “Titanic”?
- Quality: consistency
  - Users are similar if they are watching similar movies
  - Movies are similar if they are watched by similar users

Mutual clustering

- Simultaneous clustering of users and resources that users are interested in
- Quality measure: consistency between metrics on user's space and resource's space
  - Users are similar if they are interested in similar resources
  - Resources are similar if similar users are interested in this resources

Association rules mining

- How to locate best goods in supermarket?
  - Non-metrical approach, so there is no concept of distance between resources and users
- ItemSet A, AssocRule $A \rightarrow B$
  - $\mathrm{Supp}(A) = \frac{\#\{u: \forall m \in A\ u(m)=1\}}{|U|}$
  - $Conf(A \rightarrow B) = \frac{Supp(A \cup B)}{Supp(A)}$
  - $\mathrm{Supp}(A \rightarrow B) = Supp(A \cup B)$

	m1	m2	m3	m4	mK
u1	1	1	0	1	0
u2	1	1	0	0	0
u3	1	1	0	0	1
u4	1	0	0	1	0
u5	0	0	1	1	0
u6	0	0	1	1	1
m: Movies					
u: Users					

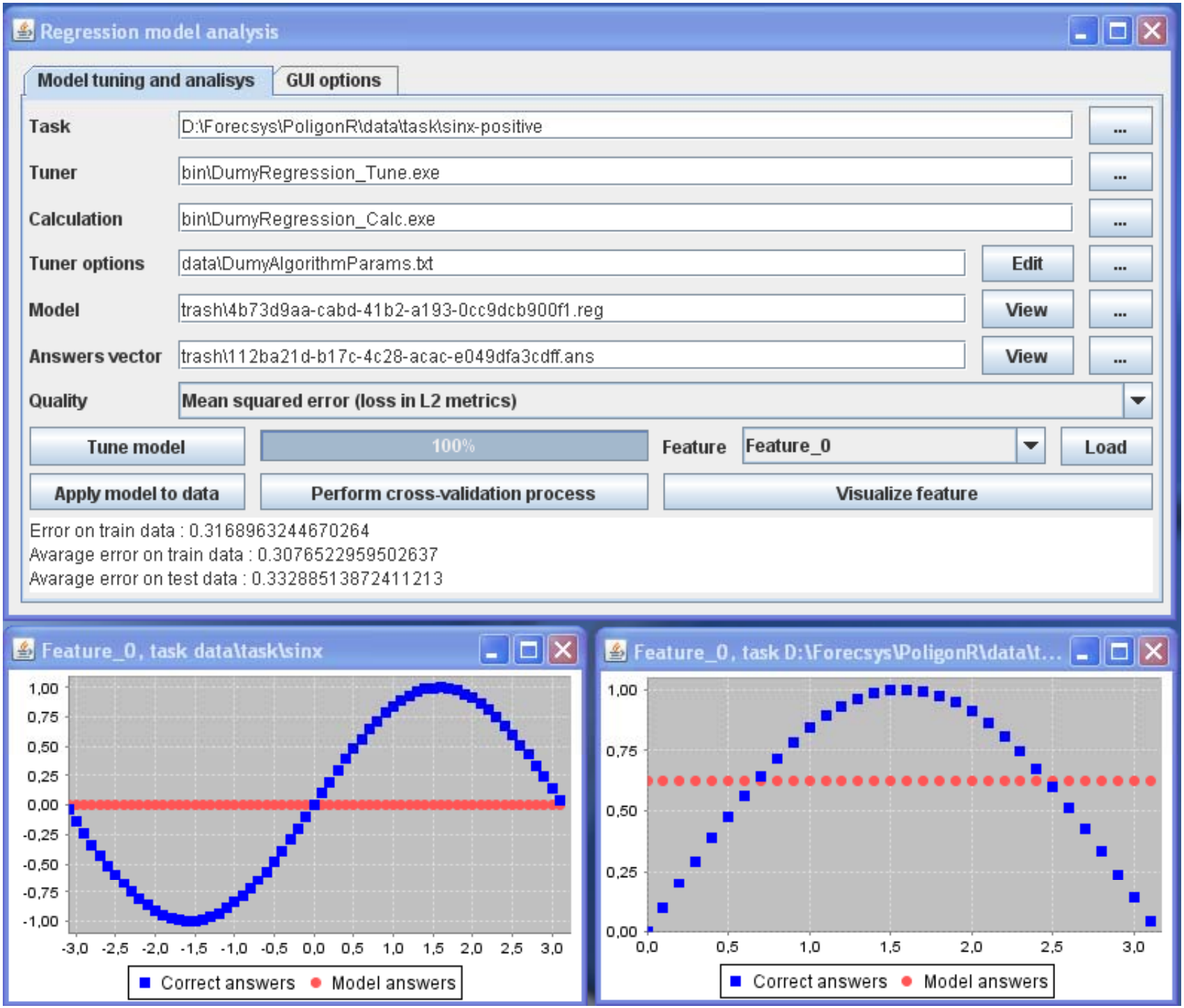
- Two quality measures for AssocRule $A \rightarrow B$.

Learning to Rank

- How to evaluate relevancy mark?
  - Precision@N / MAP / nDCG / Click through rate
- Datasets are pretty expensive
  - No UCI reposiroty
  - LETOR database from Microsoft
  - *** from Yahoo reported as 2 million USD to collect
  - TREC competition [paid, unless you are participant]
  - ***

Putting it all together:

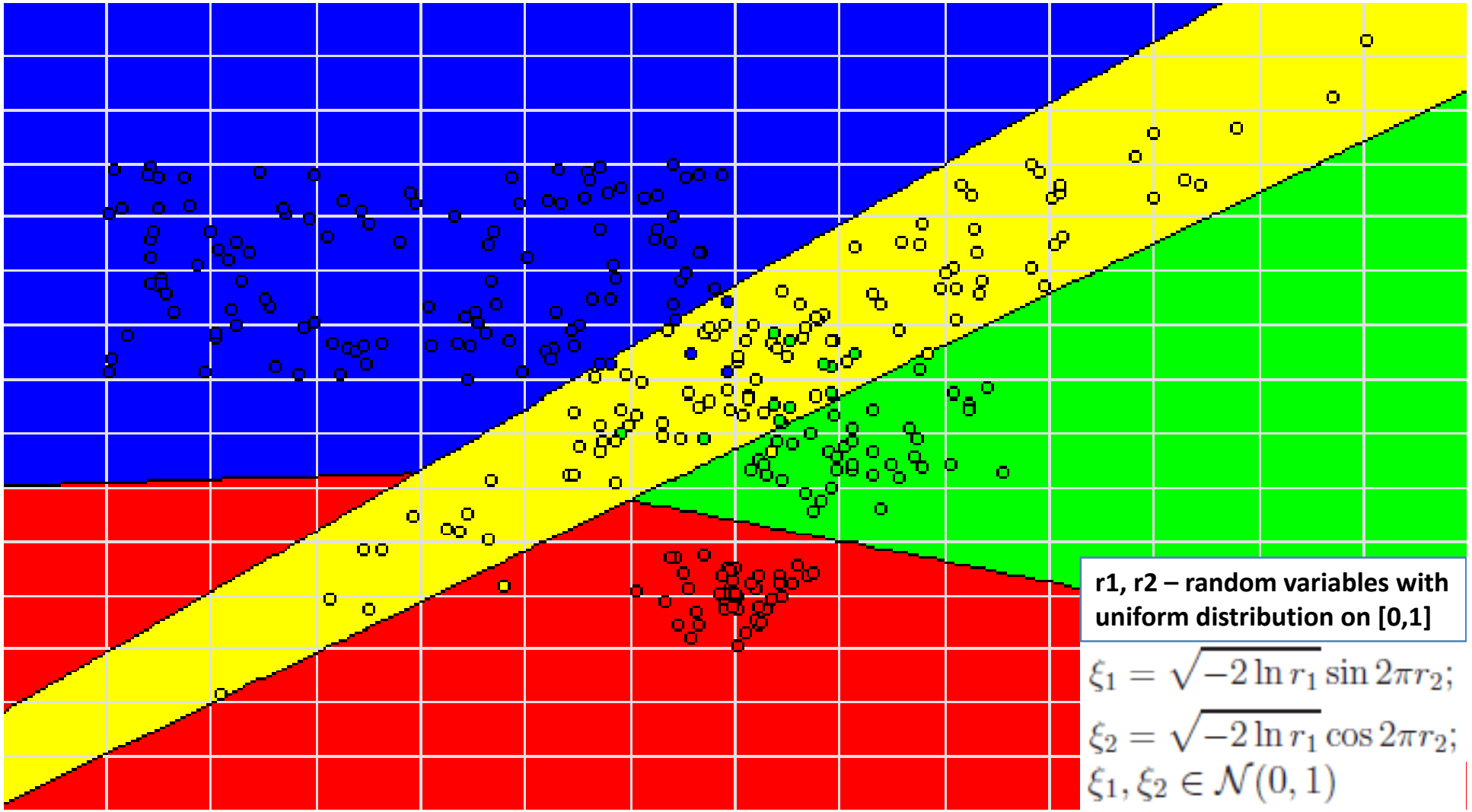
Field	Data to learn from	Example of Loss function
Classification	Items feature matrix and target classes	Weighted amount of errors
Regression		Mean square difference
Clustering	Items feature matrix, or pairwise distances	Average distance between clusters
Mutual clustering	Binary matrix of transactions	Consistency between metric on users and metric on resources
Association rules mining		Convolution of support and confidence
Learning to rank	{document, query, relevancy}	Precision@N, MAP, nDCG, Click through rate
Everywhere ☺		How much money did our prediction model save to customer? ($$$)


Regression model analysis
Model tuning and analisys
GUI options
Task
D:\Forecsys\PoligonR\data\task\sinx-positive
Tuner
bin\DumyRegression_Tune.exe
Calculation
bin\DumyRegression_Calc.exe
Tuner options
data\DumyAlgorithmParams.txt
Edit
Model
trash\4b73d9aa-cabd-41b2-a193-0cc9dcb900f1.reg
View
Answers vector
trash\112ba21d-b17c-4c28-acac-e049dfa3cdff.ans
View
Quality
Mean squared error (loss in L2 metrics)
Tune model
100%
Feature
Feature_0
Load
Apply model to data
Perform cross-validation process
Visualize feature
Error on train data : 0.3168963244670264
Avarage error on train data : 0.3076522959502637
Avarage error on test data : 0.33288513872411213
Feature_0, task data\task\sinx
1,00
0,75
0,50
0,25
0,00
-0,25
-0,50
-0,75
-1,00
-3,0 -2,5 -2,0 -1,5 -1,0 -0,5 0,0 0,5 1,0 1,5 2,0 2,5 3,0
Correct answers
Model answers
Feature_0, task D:\Forecsys\PoligonR\data\t...
1,00
0,75
0,50
0,25
0,00
0,0 0,5 1,0 1,5 2,0 2,5 3,0
Correct answers
Model answers

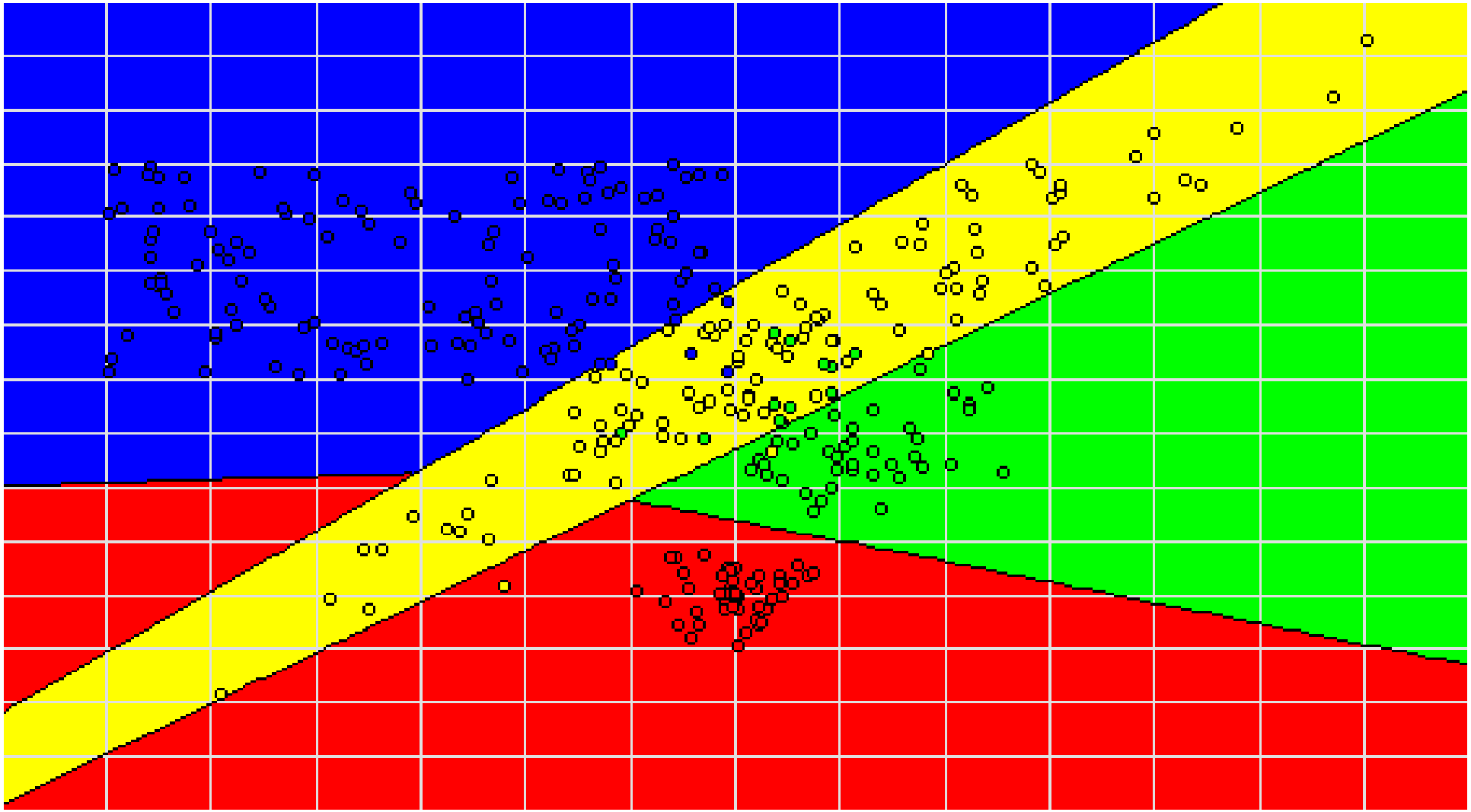
Statistical Learning Theory (SLT)

- Set of items $\mathbb{X}$ is a probability space with unknown measure distribution
- $\mathbb{F}$ – set of predictor functions to choose the best one
- $I(f,x)$ – error indicator (for predictor $f \in \mathbb{F}$ and item $x \in \mathbb{X}$)
- $\mathrm{P}(f) = E_{\mathbb{X}} I(f,x)$ – "true" error probability for $f \in \mathbb{F}$
- $\nu(f,X^\ell) = \frac{1}{\ell}\sum_{i=1}^{\ell} I(f,x_i)$ – error frequency on train data set $X^\ell = \{x_1, x_2, \ldots, x_\ell\}$.
- Is it true that $\nu(f,X^\ell) \xrightarrow{\ell\to\infty} P(f)$?

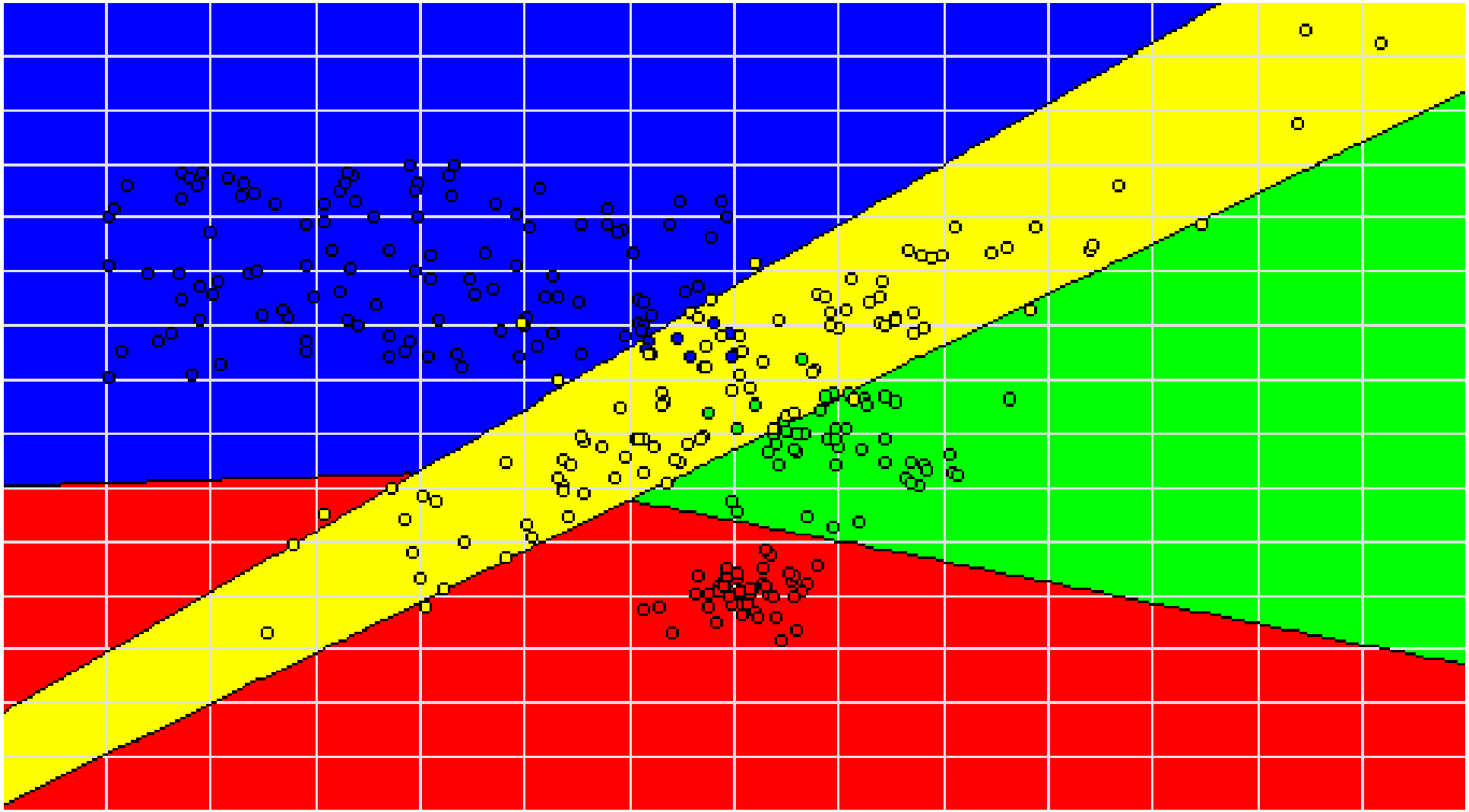
Support Vector Machine (SVM)
Train Sample



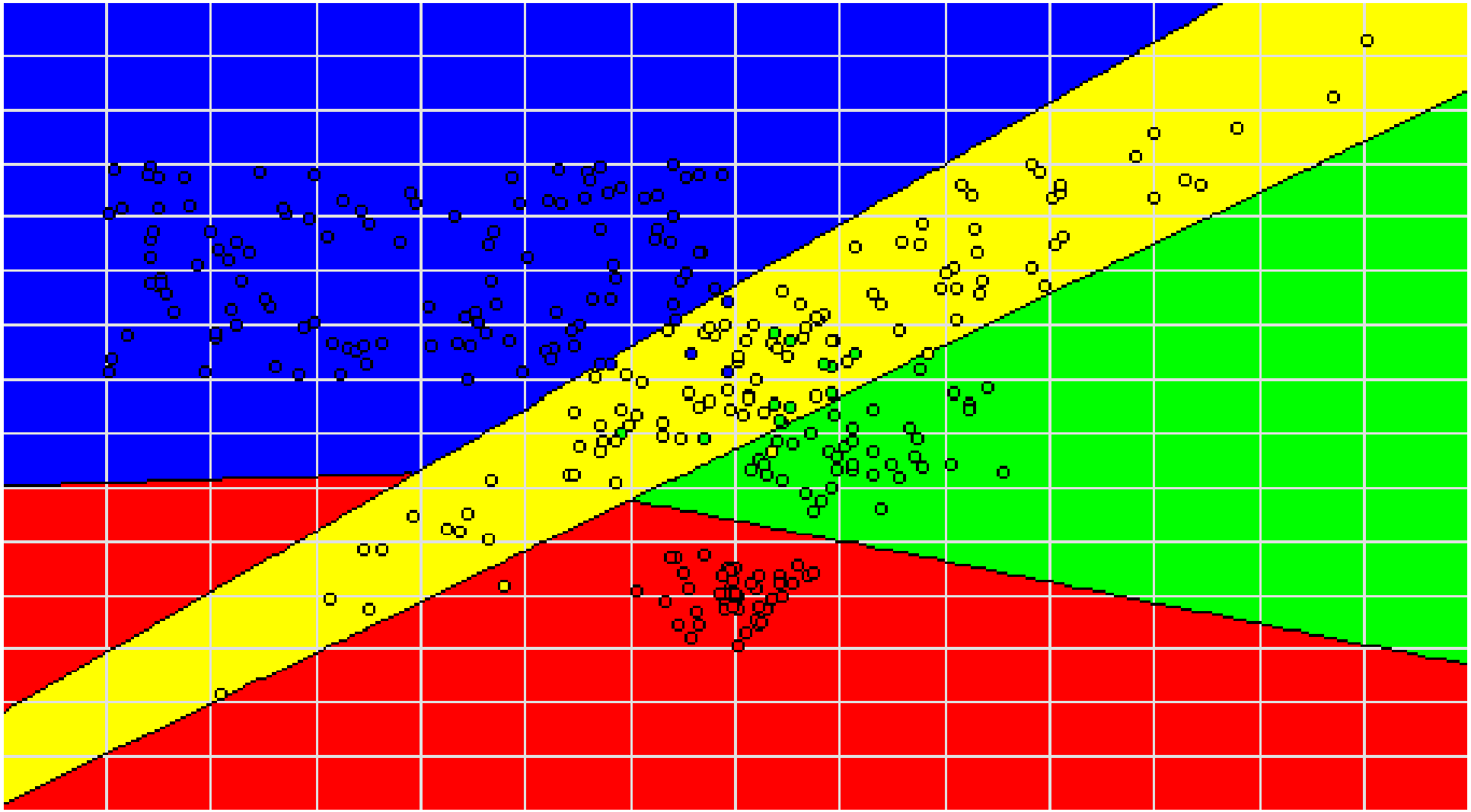
Support Vector Machine (SVM)
Train Sample



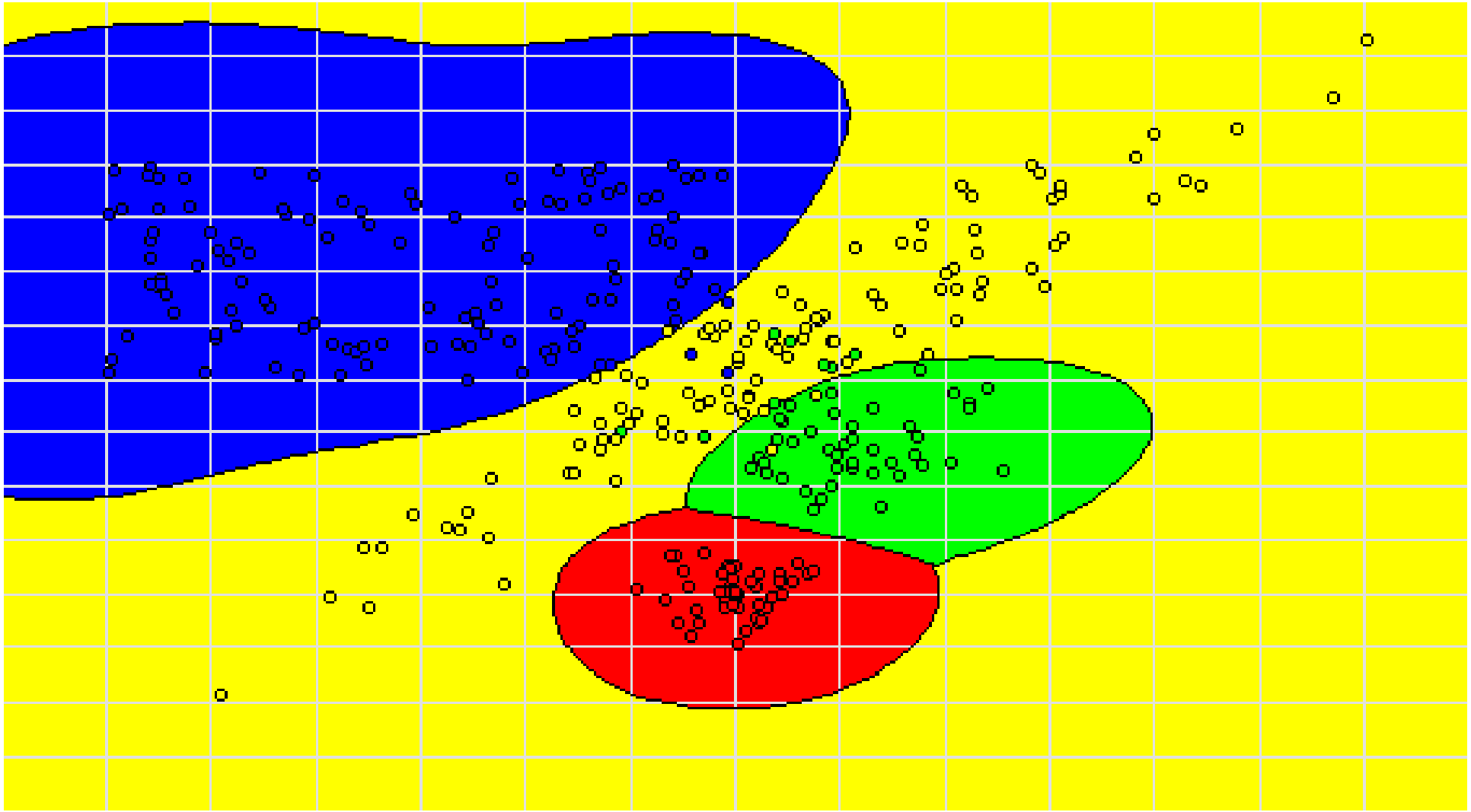
Support Vector Machine (SVM)
Test Sample



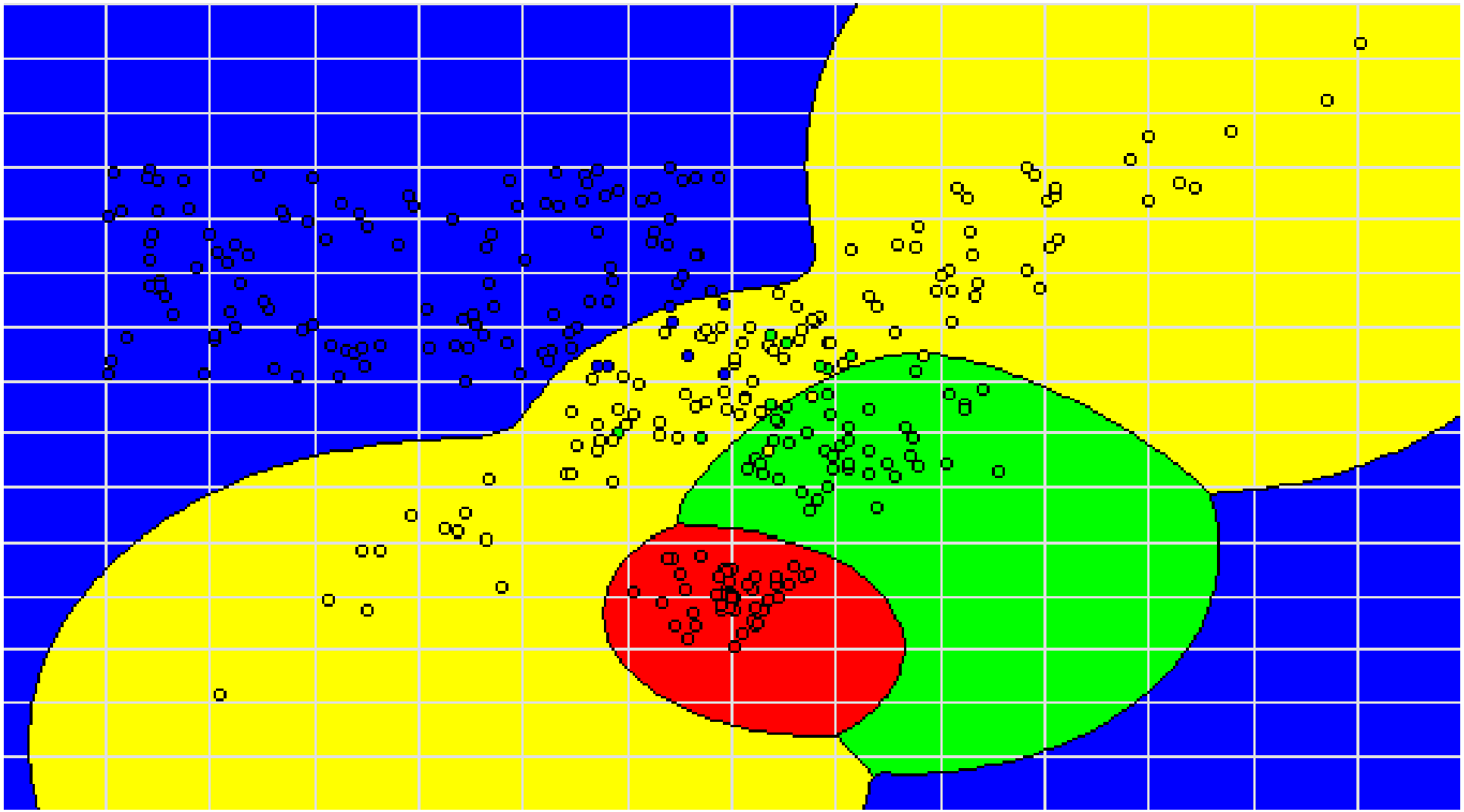
Support Vector Machine (SVM)
Train Sample



SVM with Radial Basis Functions
Train Sample



Probability distribution evaluation
Train Sample



Statistical Learning Theory (SLT)

- Set of items $\mathbb{X}$ is a probability space with unknown measure distribution $\Pr(x)$
- $\mathbb{F}$ – set of predictor functions to choose the best one
- $I(f,x)$ – error indicator (for predictor $f \in \mathbb{F}$ and item $x \in \mathbb{X}$)
- $\mathrm{P}(f) = E_{\mathbb{X}} I(f,x)$ – "true" error probability for $f \in \mathbb{F}$

$$I(f,x) \in \{0,1\} \to P(f) = \int_{x:I(f,x)=1} \Pr(x)\, dx$$

- $\nu(f,X^\ell) = \frac{1}{\ell}\sum_{i=1}^{\ell} I(f,x_i)$ – error frequency on train data set $X^\ell = \{x_1, x_2, \ldots, x_\ell\}$.
- Is it true that $\nu(f,X^\ell) \xrightarrow{\ell\to\infty} P(f)$?

Statistical Learning Theory (SLT)

- What is bad situation?
- We don't want $\left|P(f) - \nu(f, X^\ell)\right| > \epsilon$ to occur...
- ... for any predictor $f \in \mathbb{F}$:

$$\sup_{f \in \mathbb{F}} \left|P(f) - \nu(f, X^\ell)\right| > \epsilon$$

- But it happens, so probability of this event may be small at least when we have a lot of data:

$$\forall \epsilon \quad \mathrm{P}_{X^\ell} \left\{ \sup_{f \in \mathbb{F}} \left|P(f) - \nu(f, X^\ell)\right| > \epsilon \right\} \leq \eta_\ell(\epsilon) \xrightarrow{\ell \to \infty} 0$$

Statistical Learning Theory (SLT)

- Introducing "ghost" sample X^k (to have $|\mathbb{F}| \leq 2^{\ell+k}$)
- Considering only one-side deviation
- Union bound inequality

$$\mathrm{P}_{\mathbb{F}}(\epsilon) = P_{X^\ell}\left\{\sup_{f \in \mathbb{F}} \left|P(f) - \nu(f, X^\ell)\right| > \epsilon\right\}$$

$$\leq 2\, P_{X^k, X^\ell}\left\{\max_{f \in \mathbb{F}} \left|\nu(f, X^k) - \nu(f, X^\ell)\right| > \frac{\epsilon}{2}\right\}$$

$$\leq P_{X^k, X^\ell}\left\{\max_{f \in \mathbb{F}} \left(\nu(f, X^k) - \nu(f, X^\ell)\right) > \frac{\epsilon}{2}\right\}$$

$$\leq \sum_{f \in \mathbb{F}} P_{X^k, X^\ell}\left\{\nu(f, X^k) - \nu(f, X^\ell) > \frac{\epsilon}{2}\right\}$$

Statistical Learning Theory (SLT)

- Last steps:

$$\mathrm{P}_{X^\ell}\left\{\sup_{f\in\mathbb{F}}\left|P(f)-\nu(f,X^\ell)\right|>\epsilon\right\}$$

$$\leq \sum_{f\in\mathbb{F}} P_{X^k,X^\ell}\left\{\nu(f,X^k)-\nu(f,X^\ell)>\frac{\epsilon}{2}\right\}$$

- Few useful definitions:
  - Error frequency deviation on train and test datasets:

$$\delta(f,X^\ell,X^k)=\nu(f,X^k)-\nu(f,X^\ell)$$

  - Probability of ovefitting:

$$\mathrm{P}_{\mathbb{F}}(\epsilon)=\mathrm{P}_{X^k,X^\ell}\left\{\max_{f\in\mathbb{F}}\delta(f,X^\ell,X^k)>\epsilon\right\}$$

Statistical Learning Theory (SLT)

- Last step: bounded differences inequality (McDiarmid)

$$\mathrm{P}_{\mathbb{F}}(\epsilon) = \mathrm{P}_{X^\ell}\left\{\sup_{f \in \mathbb{F}} \left|P(f) - \nu(f, X^\ell)\right| > \epsilon\right\}$$

$$\leq \sum_{f \in \mathbb{F}} P_{X^k, X^\ell}\left\{\nu(f, X^k) - \nu(f, X^\ell) > \frac{\epsilon}{2}\right\}$$

$$\leq |\mathbb{F}| \times \frac{3}{2} e^{-\epsilon^2 \ell}$$

Statistical Learning Theory (SLT)

- VC-bound:
  - Let $f_1 \sim f_2$ when $I(f_1, x) = I(f_2, x)$ holds for all $x \in X^L$
  - Let $\Delta^{\mathbb{F}}(X^L)$ be an amount of equivalence classes on $\mathbb{F}$, induced by dataset X^L.
  - Let $\Delta^{\mathbb{F}}(L) = \sup_{X^L} \Delta^{\mathbb{F}}(X^L)$.

  Then

$$\mathrm{P}_{\mathbb{F}}(\epsilon) \leq \Delta^{\mathbb{F}}(2\ell) \cdot \frac{3}{2} e^{-\epsilon^2 \ell} = \eta_{VC}(\ell, \epsilon).$$

- Trivial estimation: $\Delta^{\mathbb{F}}(L) \leq 2^L$
- Example: linear classifier built on h features
  - Finite capacity: $\Delta^{\mathbb{F}}(L) \leq \frac{3}{2} \frac{L^h}{h!}$, so polynomial on h.
  - And then we have learnability, since

$$\forall \epsilon > 0 \quad \eta_{VC}(\ell, \epsilon) \to 0, \ell \to \infty.$$

Statistical Learning Theory (SLT)

- Philosophy: Occam's razor
  - "The simplest explanation is more likely the correct one"
  - "Entities must not be multiplied beyond necessity"
- Math language: for all $f \in \mathbb{F}$ with probability $1 - \eta$

$$\nu(f, X^k) \leq \nu(f, X^\ell) + \sqrt{\frac{h}{\ell} \ln\left(\frac{2e\ell}{h}\right) + \frac{4}{9\ell} ln \frac{1}{\eta}}$$

- Complexity penalty $\mathrm{CP}(\eta, h, \ell)$, or regularization.
  - Set of nested model sets with increasing complexity:

$$\mathbb{F}_1 \subset \mathbb{F}_2 \subset \ldots \subset \mathbb{F}_h$$

Problems of Statistical Learning Theory

- Very week upper bounds,
  - and no way to measure where did we loose precision:

$$\mathrm{P}_{X^\ell}\left\{\sup_{f\in\mathbb{F}}\left|P(f)-\nu\left(f,X^\ell\right)\right|>\epsilon\right\}\leq\eta_\ell\left(\epsilon\right)\overset{\ell\to\infty}{\longrightarrow}0$$

- No concept of learning method

$$\cancel{\mu\left(X^\ell\right)=f}\qquad\sup_{f\in\mathbb{F}}$$

- Union bound: $\sum_{f\in\mathbb{F}}\mathrm{P}_X\{\left|P(f)-\nu\left(f,X^\ell\right)\right|>\epsilon\}$
- Data worst-case: $\Delta^{\mathbb{F}}(L)=\sup_{X^L}\Delta^{\mathbb{F}}(X^L)$

Problem of Statistical Learning Theory

Too far away from reality!

Combinatorial Learning Theory

New starting point:

- Learning method: $f = \mu(X^\ell)$
- Forget about probability distribution over $\mathbb{X}$:

$$\delta_\mu(X^\ell, X^k) = \nu\big(\mu(X^\ell), X^k\big) - \nu\big(\mu(X^\ell), X^\ell\big)$$

- Cross-validation:

$$\mathrm{P}_\mu(\epsilon) = \frac{1}{\binom{L}{\ell}} \sum_{X^L = X^\ell \sqcup X^k} [\delta_\mu(X^\ell, X^k) > \epsilon]$$

Combinatorial Learning Theory

New starting point:

- Learning method: $f = \mu(X^\ell)$
- Forget about probability distribution over $\mathbb{X}$:

$$\delta_\mu(X^\ell, X^k) = \nu\big(\mu(X^\ell), X^k\big) - \nu\big(\mu(X^\ell), X^\ell\big)$$

- Cross-validation:

$$\mathrm{P}_\mu(\epsilon) = \underbrace{\frac{1}{\binom{L}{\ell}} \sum_{X^L = X^\ell \sqcup X^k}}_{\mathbf{P}} [\delta_\mu(X^\ell, X^k) > \epsilon]$$

Combinatorial Learning Theory

New starting point:

- ~~Learning method: $f = \mu(X^\ell)$~~
- Forget about probability distribution over $\mathbb{X}$:

$$\delta(X^\ell, X^k) = \sup_{f \in \mathbb{F}} \left(\nu(f, X^k) - \nu(f, X^\ell)\right)$$

- Cross-validation:

$$\mathrm{P}_{\mathbb{F}}(\epsilon) = \frac{1}{\binom{L}{\ell}} \sum_{X^L = X^\ell \sqcup X^k}^{L} [\delta(X^\ell, X^k) > \epsilon]$$

Combinatorial Learning Theory

New starting point:

- ~~Learning method: $f = \mu(X^\ell)$~~
- ~~Forget about probability distribution over $\mathbb{X}$:~~

$$\delta(X^\ell, X^k) = \sup_{f \in \mathbb{F}} \left(P_{\mathbb{X}}(f) - \nu(f, X^\ell) \right)$$

- Cross-validation:

$$\mathrm{P}_{\mathbb{F}}(\epsilon) = \frac{1}{\binom{L}{\ell}} \sum_{X^L = X^\ell \sqcup X^k} [\delta(X^\ell, X^k) > \epsilon]$$

Combinatorial Learning Theory

New starting point:

- ~~Learning method:~~ $f = \mu(X^\ell)$
- ~~Forget about probability distribution over~~ $\mathbb{X}$:

$$\delta(X^\ell) = \sup_{f \in \mathbb{F}} \left(P_{\mathbb{X}}(f) - \nu(f, X^\ell) \right)$$

- ~~Cross-validation:~~

$$\mathrm{P}_{\mathbb{F}}(\epsilon) = \mathrm{P}_{X^\ell}\{\delta(X^\ell) > \epsilon\}$$

Combinatorial Learning Theory

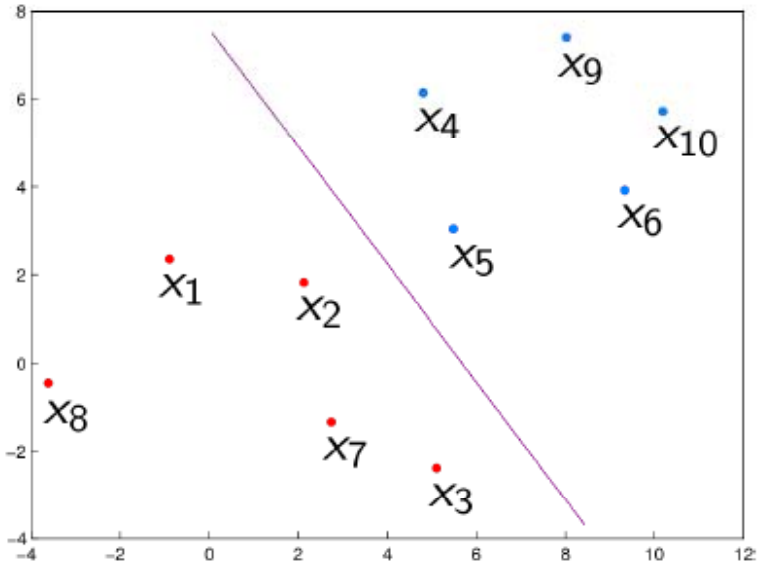
- Let $\mathrm{X}^L = \{(x_i, y_i)\}_{i=1}^L$ – dataset, $L = \ell + k$
- Let $\mathbb{F}$ – set of predictor functions
- $I(f, x)$ – error indicator
- Error rate: $\nu(f, U) = |\{x \in U : I(f, x) = 1\}|/|U|$
- Learning method: mapping $\mu: 2^{\mathrm{X}^L} \to \mathbb{F}$
- Overfitting: $\nu\left(\mu(X^\ell), X^k\right) - \nu\left(\mu(X^\ell), X^\ell\right) > \epsilon$
- Probability of overfitting:

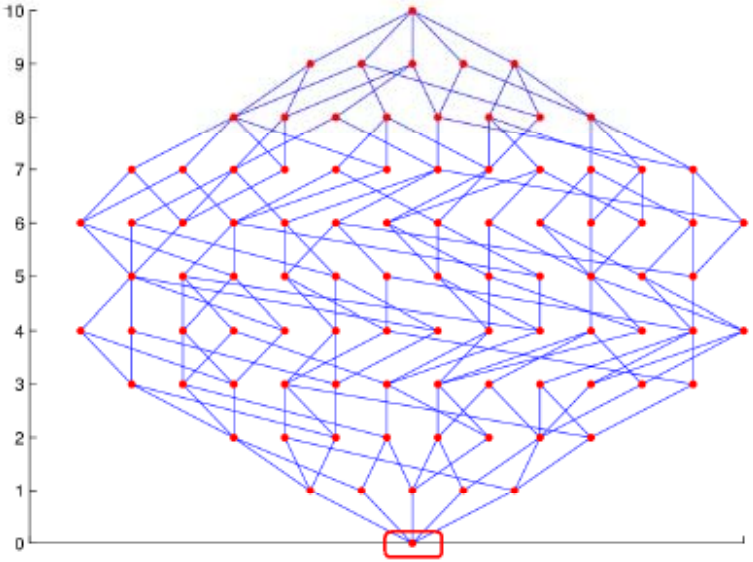
$$\mathrm{P}_{\mathbb{F}}(\epsilon) = \underbrace{\frac{1}{\binom{L}{\ell}} \sum_{X^L = X^\ell \sqcup X^k}^{L}}_{\mathbf{P}} \left[\nu\left(\mu(X^\ell), X^k\right) - \nu\left(\mu(X^\ell), X^\ell\right) > \epsilon\right]$$

Combinatorial Learning Theory

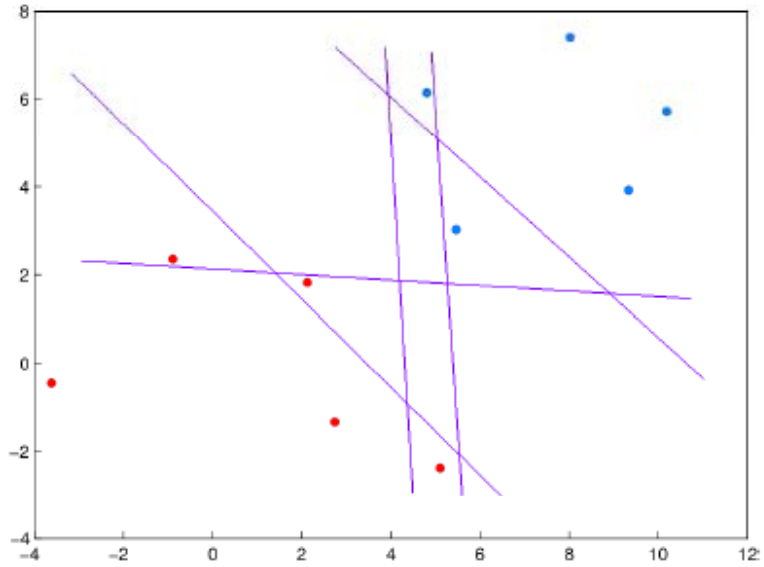
Learning method: Empirical Risk Minimization

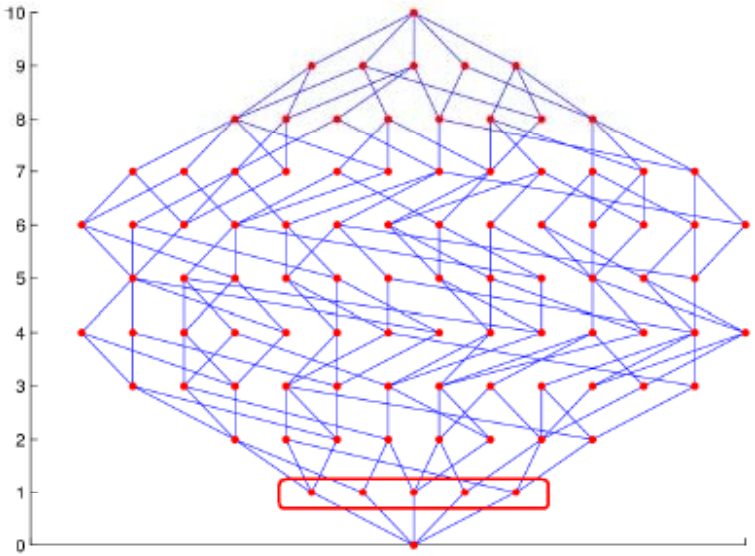
	f_1	f_2	f_3	f_4	f_5	f_6	$\cdots$	f_D
x_1	1	1	0	0	0	1	$\cdots$	1
$\ldots$	0	0	0	0	1	1	$\cdots$	1
x_ℓ	0	0	1	0	0	0	$\cdots$	0
$x_{\ell+1}$	0	0	0	1	1	1	$\cdots$	0
$\ldots$	0	0	0	1	0	0	$\cdots$	1
x_L	0	1	1	1	1	1	$\cdots$	0



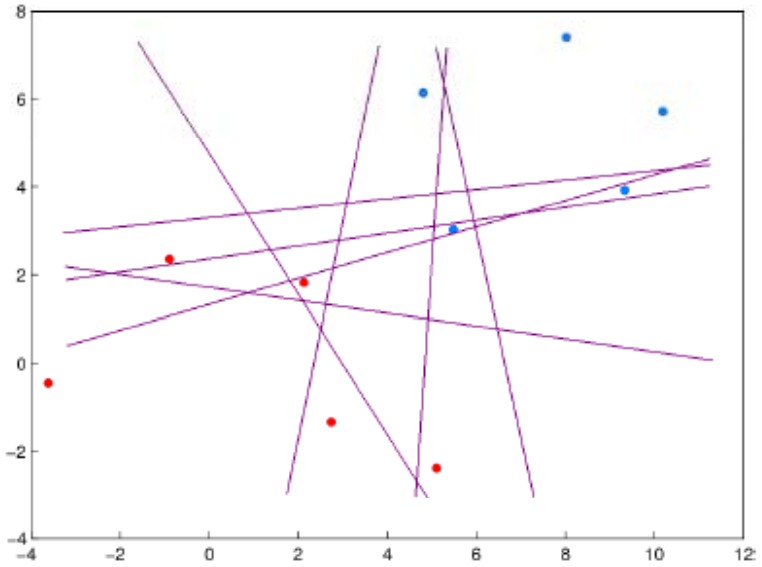


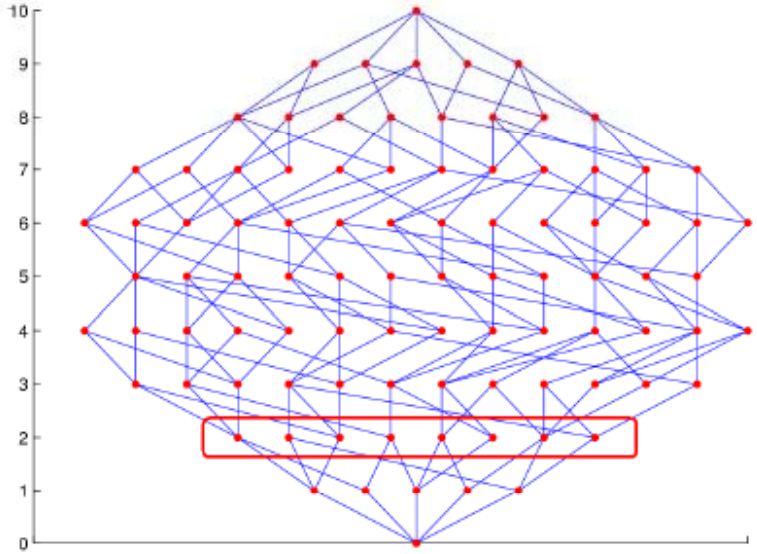
x_1	0
x_2	0
x_3	0
x_4	0
x_5	0
x_6	0
x_7	0
x_8	0
x_9	0
x_{10}	0





x_1	0	1	0	0	0	0
x_2	0	0	1	0	0	0
x_3	0	0	0	1	0	0
x_4	0	0	0	0	1	0
x_5	0	0	0	0	0	1
x_6	0	0	0	0	0	0
x_7	0	0	0	0	0	0
x_8	0	0	0	0	0	0
x_9	0	0	0	0	0	0
x_{10}	0	0	0	0	0	0

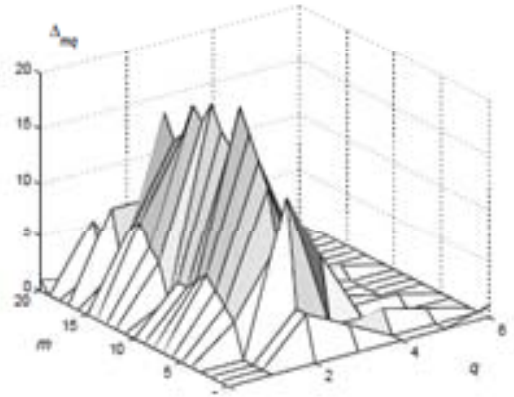




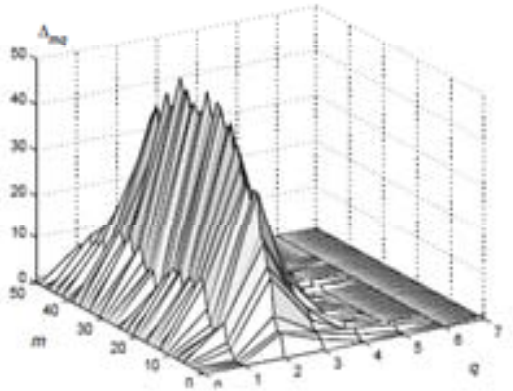
x_1	0	1	0	0	0	0	1	0	0	0	0	1	1	0	. . .
x_2	0	0	1	0	0	0	1	1	0	0	0	0	0	0	. . .
x_3	0	0	0	1	0	0	0	1	1	0	0	0	0	1	. . .
x_4	0	0	0	0	1	0	0	0	1	1	0	0	0	0	. . .
x_5	0	0	0	0	0	1	0	0	0	1	1	1	0	0	. . .
x_6	0	0	0	0	0	0	0	0	0	0	1	0	1	0	. . .
x_7	0	0	0	0	0	0	0	0	0	0	0	0	0	1	. . .
x_8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	. . .
x_9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	. . .
x_{10}	0	0	0	0	0	0	0	0	0	0	0	0	0	0	. . .

Attributes of connectivity graph

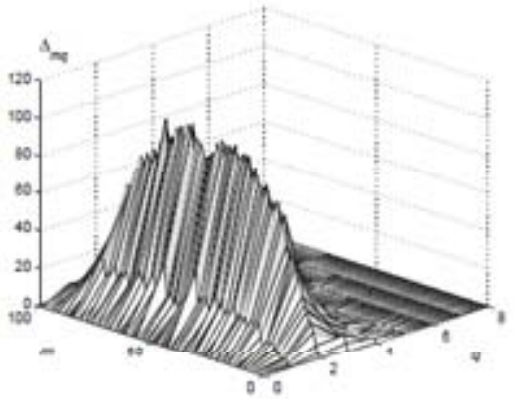
$L = 20$



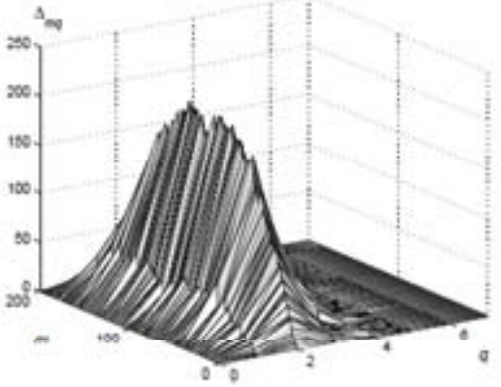
$L = 50$



$L = 100$



$L = 200$



$D(m, q)$ - how many predictors with m errors do we have, such that each of them is "connected" with exact q predictors with $m + 1$ errors.

Combinatorial Learning Theory

- Empirical risk minimization:

$$\mu(X^\ell) = \underset{f \in \mathbb{F}}{\operatorname{argmin}}\, \nu(f, X^\ell)$$

- Pessimistic ERM:

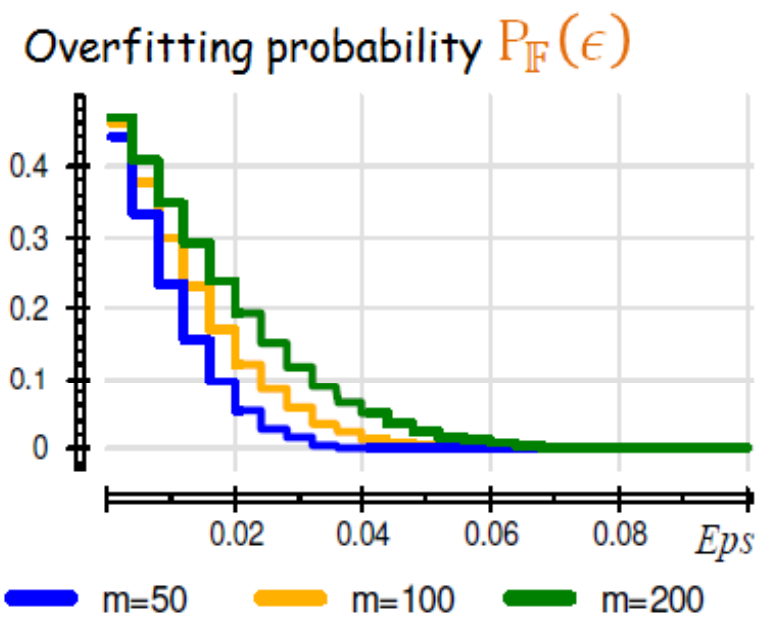
$$\mathbb{F}(X^\ell) = \underset{f \in \mathbb{F}}{\operatorname{Argmin}}\, \nu(f, X^\ell),$$

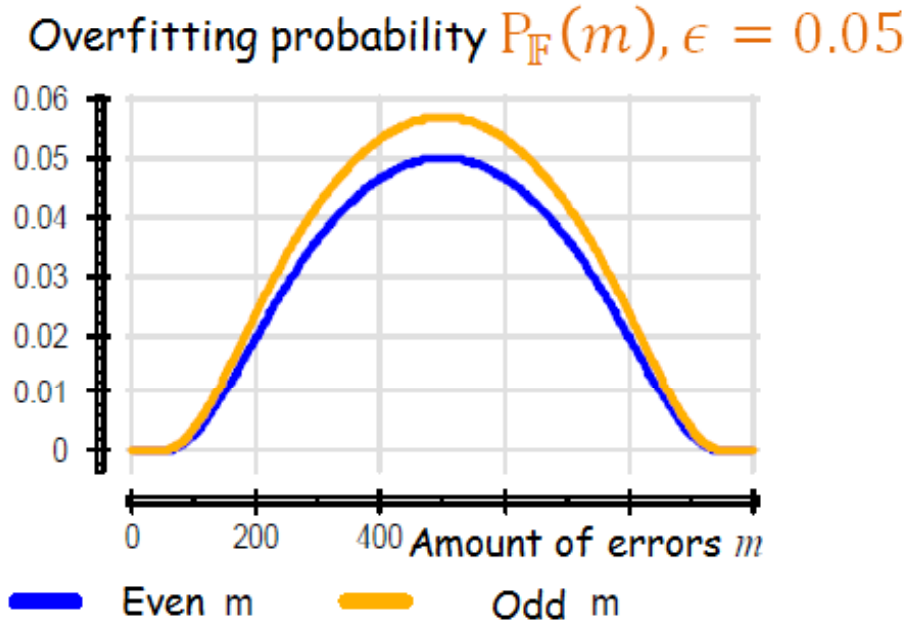
$$\mu(X^\ell) = \underset{f \in (X^\ell)}{\operatorname{argmax}}\, \nu(f, X^k)$$

- Randomized ERM

$$\mu(X^\ell) = \frac{[f \in \mathbb{F}(X^\ell)]}{|\mathbb{F}(X^\ell)|}$$

Overfitting probability for fixed predictor



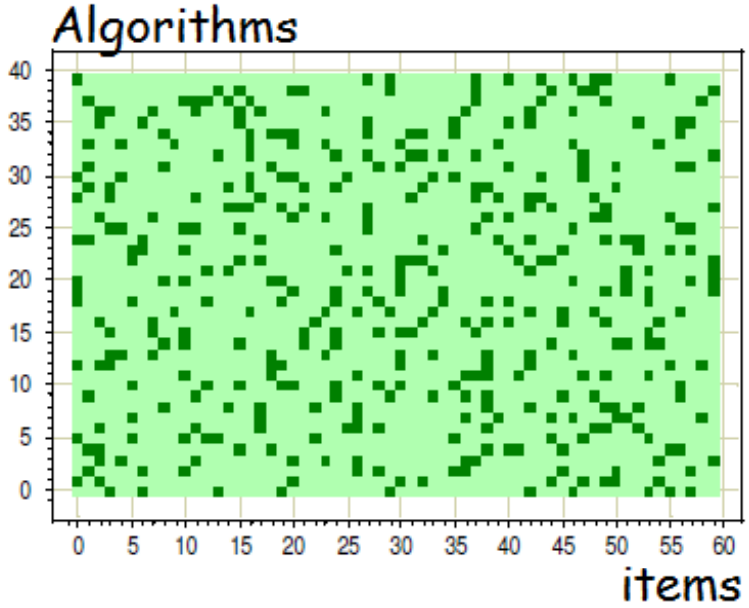


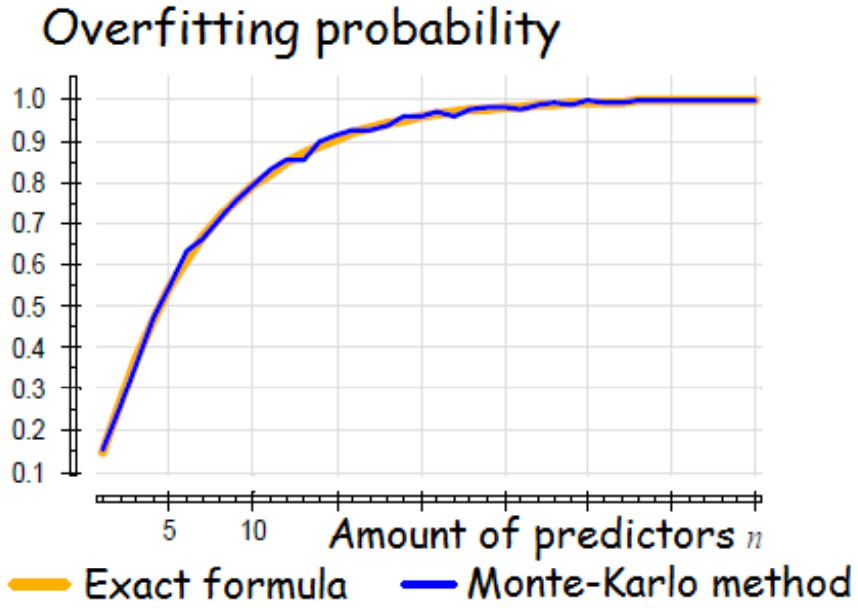
Theorem (Overfitting probability for fixed predictor)

$$P_{\mathbb{F}}(\varepsilon) = \boldsymbol{P}\Big\{\delta_\mu(X^\ell, X^k) \geq \varepsilon\Big\} = H_L^{\ell,m}(s_0),$$

where $m = n(f, \mathbb{X})$, $s_0 = \frac{\ell}{L}(m - \varepsilon k)$, $H_L^{\ell,m}(s_0) = \sum\limits_{s=0}^{\lfloor z \rfloor} \frac{C_m^s C_{L-m}^{\ell-s}}{C_L^\ell}$.

Overfitting probability: predictors with random errors





- A_m^n — set of n predictors, with m errors for each one. Errors are not correlated.

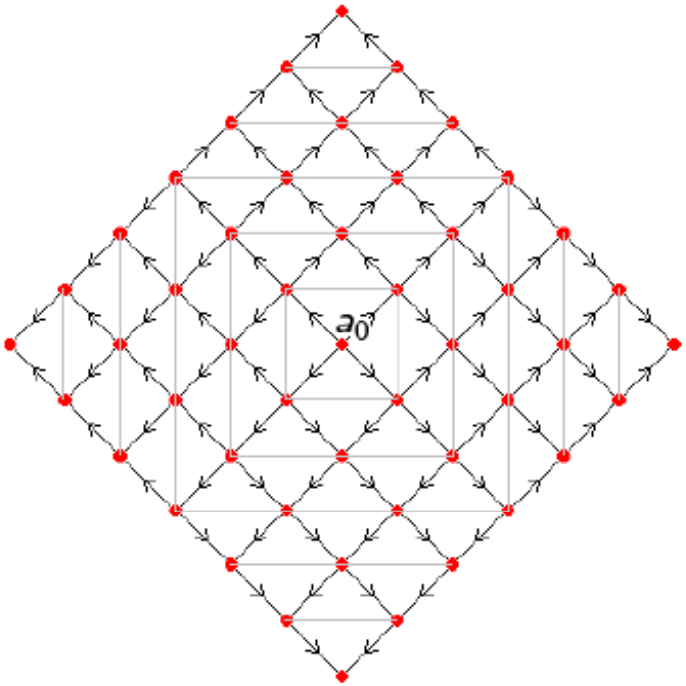
Теорема (Overfitting probability for A_m^n)

Let μ — randomized ERM. Then

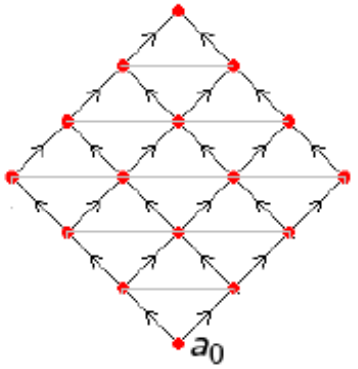
$$\boldsymbol{E}_G P_{\mathbb{F}}(\varepsilon, A_m^n) = 1 - (1 - P_{\mathbb{F}}(\varepsilon, a_m))^n$$

Continuous predictors set

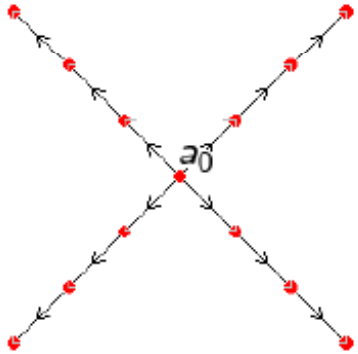
Let us study behavior of the following predictors set:



Unimodal h-dim lattice



Monotonic h-dim lattice



Monotonic chains binding

- A_B — Monotonic chains binding of h, length D,
- A_M — Monotonic h-dim lattice,
- A_U — Unimodal h-dim lattice.

Theorem (Overfitting probability A_B, A_M, and A_U.)

$$P_{\mathbb{F}}(\varepsilon, A_B) = \sum_{p=0}^{D} \sum_{S=p}^{hD} \sum_{F=0}^{h} \frac{|\omega_p| R^p_{D,h}(S,F)}{1+S} \frac{C^{\ell'}_{L'}}{C^{\ell}_{L}} H^{\ell',m}_{L'}(s_0),$$

$$P_{\mathbb{F}}(\varepsilon, A_M) = \sum_{\vec{\lambda} \in Y^{h,D}_*} \sum_{\substack{\vec{t} \geq \vec{\lambda}, \\ \|\vec{t}\| \leq D}} \frac{|S_h \vec{\lambda}|}{T(\vec{t})} \frac{C^{\ell'}_{L'}}{C^{\ell}_{L}} H^{\ell',m}_{L'}(s_0),$$

$$P_{\mathbb{F}}(\varepsilon, A_U) = \sum_{\vec{\lambda} \in Y^{h,D}_*} \sum_{\substack{\vec{t} \geq \vec{\lambda}, \\ \|\vec{t}\| \leq D}} \sum_{\substack{\vec{t'} \geq \vec{0}, \\ \|\vec{t'}\| \leq D}} \frac{|S_h \vec{\lambda}| \cdot 2^{n(\vec{\lambda})}}{T(\vec{t} + \vec{t'})} \frac{C^{\ell'}_{L'}}{C^{\ell}_{L}} H^{\ell',m}_{L'}(s_0),$$

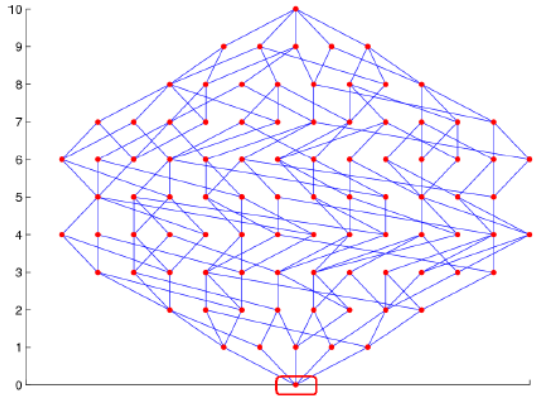
where $H^{\ell',m}_{L'}(s_0)$ *— hypergeometric distribution.*

Сравнение с классическими оценками

Оценка для одного алгоритма:

$$Q_\varepsilon = H_L^{\ell,m}\left(s_m(\varepsilon)\right).$$

Оценка Вапника-Червоненкиса (1971):



$$\vdots \leqslant \sum_{m=\lceil \varepsilon k \rceil}^{L} \Delta_m \cdot H_L^{\ell,m}\left(s_m(\varepsilon)\right) \leqslant$$

$$\leqslant |A| \cdot \max_{m=0,\ldots,L} H_L^{\ell,m}\left(s_m(\varepsilon)\right).$$

Оценка с учётом расслоения–связности (2010):

$$Q_\varepsilon \leqslant \sum_{m=\lceil \varepsilon k \rceil}^{L} \sum_{q=0}^{L} \Delta_{mq} \cdot \frac{C_{L-q}^{\ell-q}}{C_L^\ell} \cdot H_{L-q}^{\ell-q,m}\left(s_m(\varepsilon)\right).$$

Насколько важно учитывать эффекты расслоения и связности?

Эксперимент с цепочками алгоритмов:

Цепочка с расслоением:

	a_1	a_2	a_3	a_4	a_5	$\ldots$	a_D
x_1	1	1	1	1	1	1	1
x_2	0 $\rightarrow$	1	1	1	1	1	1
x_3	0	0 $\rightarrow$	1	1	1	1	1
x_4	0	0	0 $\rightarrow$	1	1	1	1
x_5	0	0	0	0 $\rightarrow$	1	1	1
x_6	0	0	0	0	0 $\rightarrow$	1	1

Цепочка без расслоения:

	a_1	a_2	a_3	a_4	a_5	$\ldots$	a_D
x_1	1	1 $\rightarrow$	0	0	0	0	0
x_2	0 $\rightarrow$	1	1	1 $\rightarrow$	0	0	0
x_3	0	0	0 $\rightarrow$	1	1	1 $\rightarrow$	0
x_4	0	0	0	0	0 $\rightarrow$	1	1
x_5	0	0	0	0	0	0	0
x_6	0	0	0	0	0	0	0

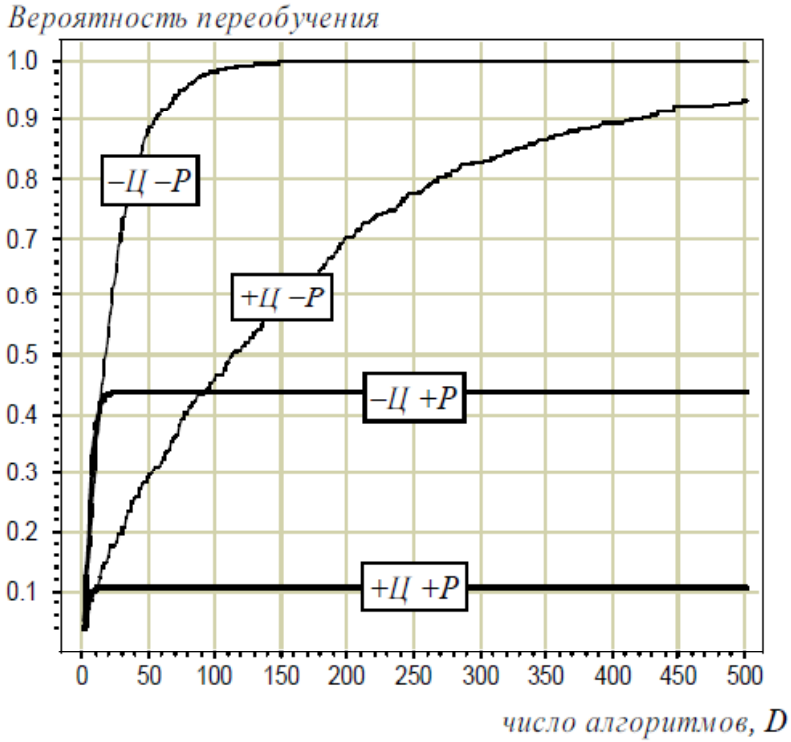
Для каждой цепочки генерируется *не-цепочка* путём случайной перестановки в каждом столбце.

Итого имеем 4 модельных семейства:

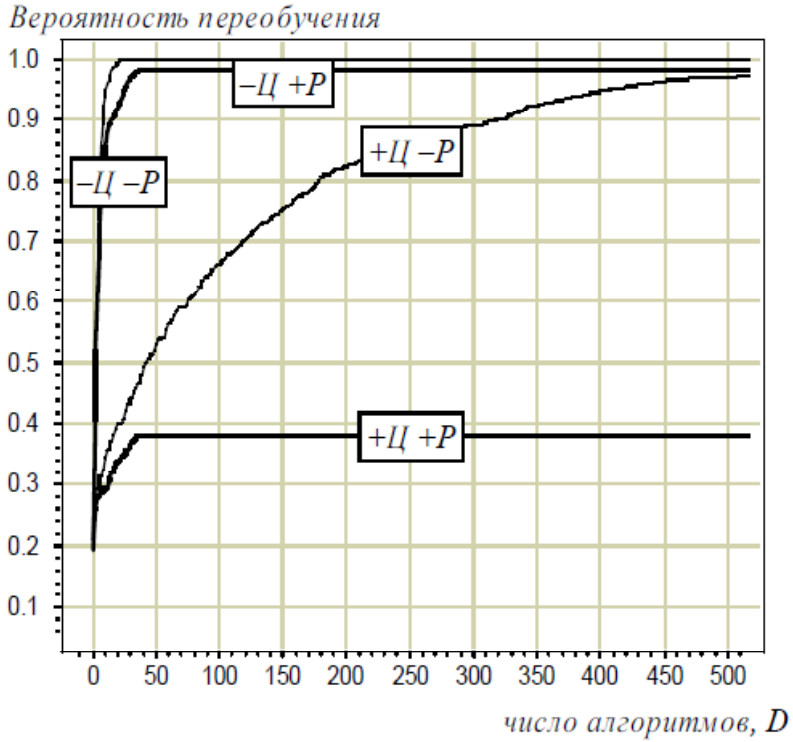
+Ц+Р	+Ц−Р
−Ц+Р	−Ц−Р

Эксперимент: зависимость Q_ε от D при $\ell = k = 100$, $\varepsilon = 0.05$

Простая задача, $n(a_1, \mathbb{X}^L) = 10$



Трудная задача, $n(a_1, \mathbb{X}^L) = 50$

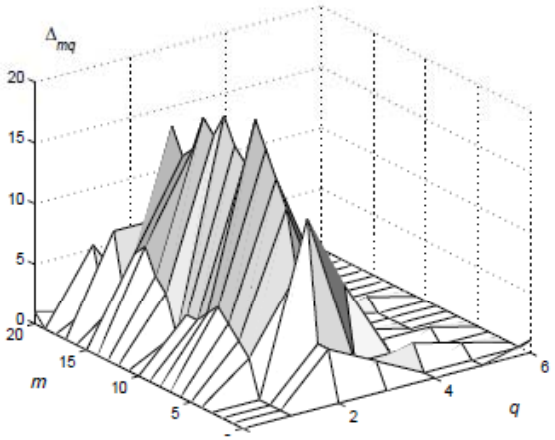


- *Связность* приводит к замедлению роста $Q_\varepsilon(D)$.
- *Расслоение* понижает уровень горизонтальной асимптоты.

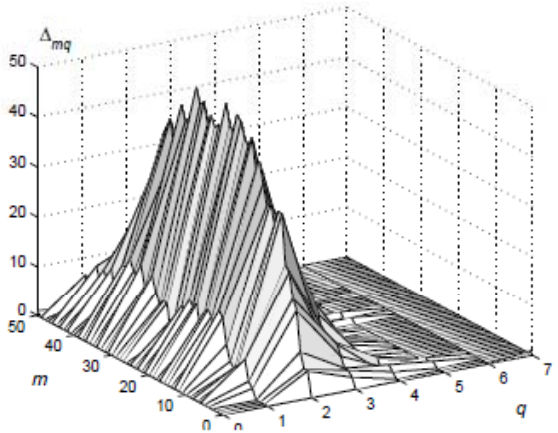
Пример. Профили расслоения-связности Δ_{mq}

линейно разделимые двумерные выборки длины L; линейные классификаторы

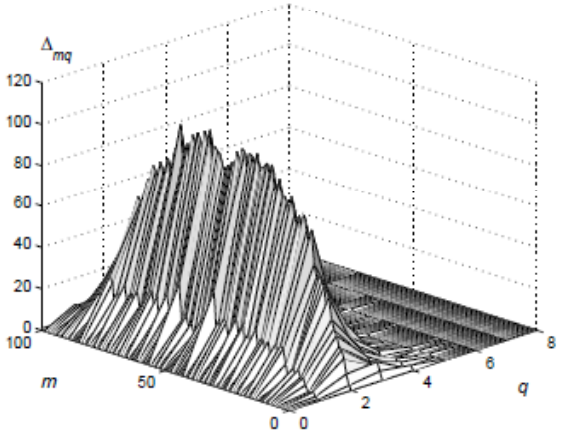
$L = 20$



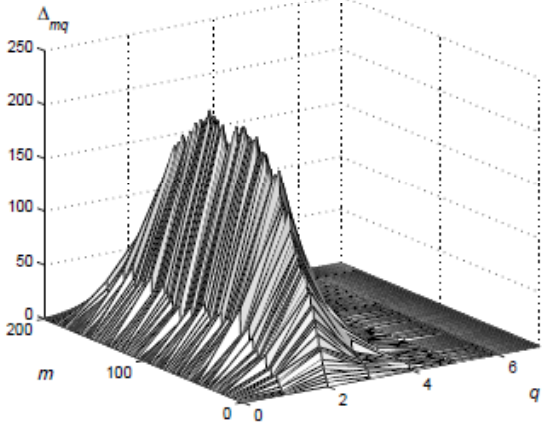
$L = 50$



$L = 100$



$L = 200$



Гипотеза сепарабельности: $\Delta_{mq} \approx \Delta_m \lambda_q$.

Гипотеза размерности: средняя связность $\approx$ размерность пространства

Conclusions about Combinatorial Learning Theory

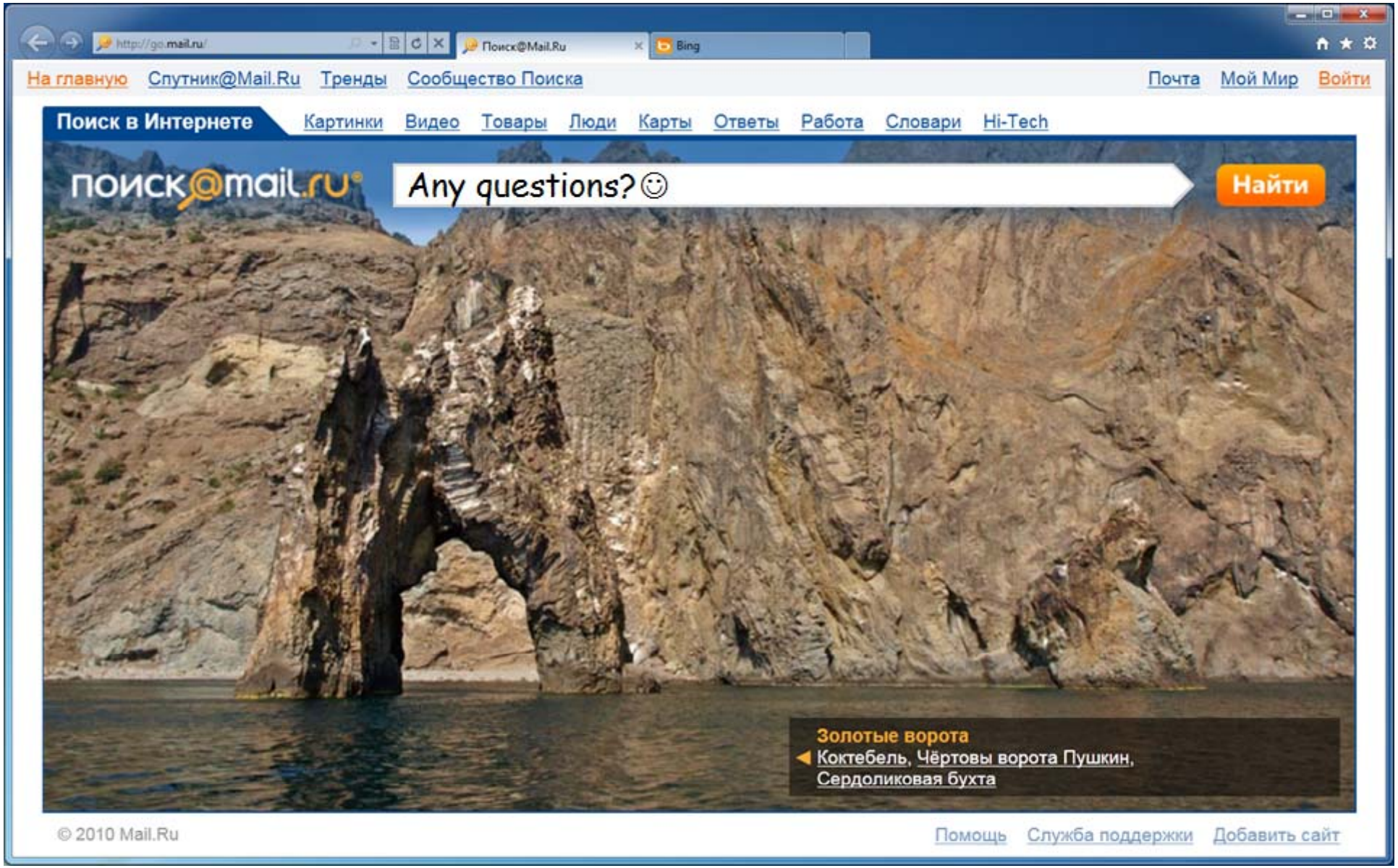
- Exact overfitting probability estimation,
  - Or tight bounds with a way to measure where did we loose precision:
- Concept of learning method

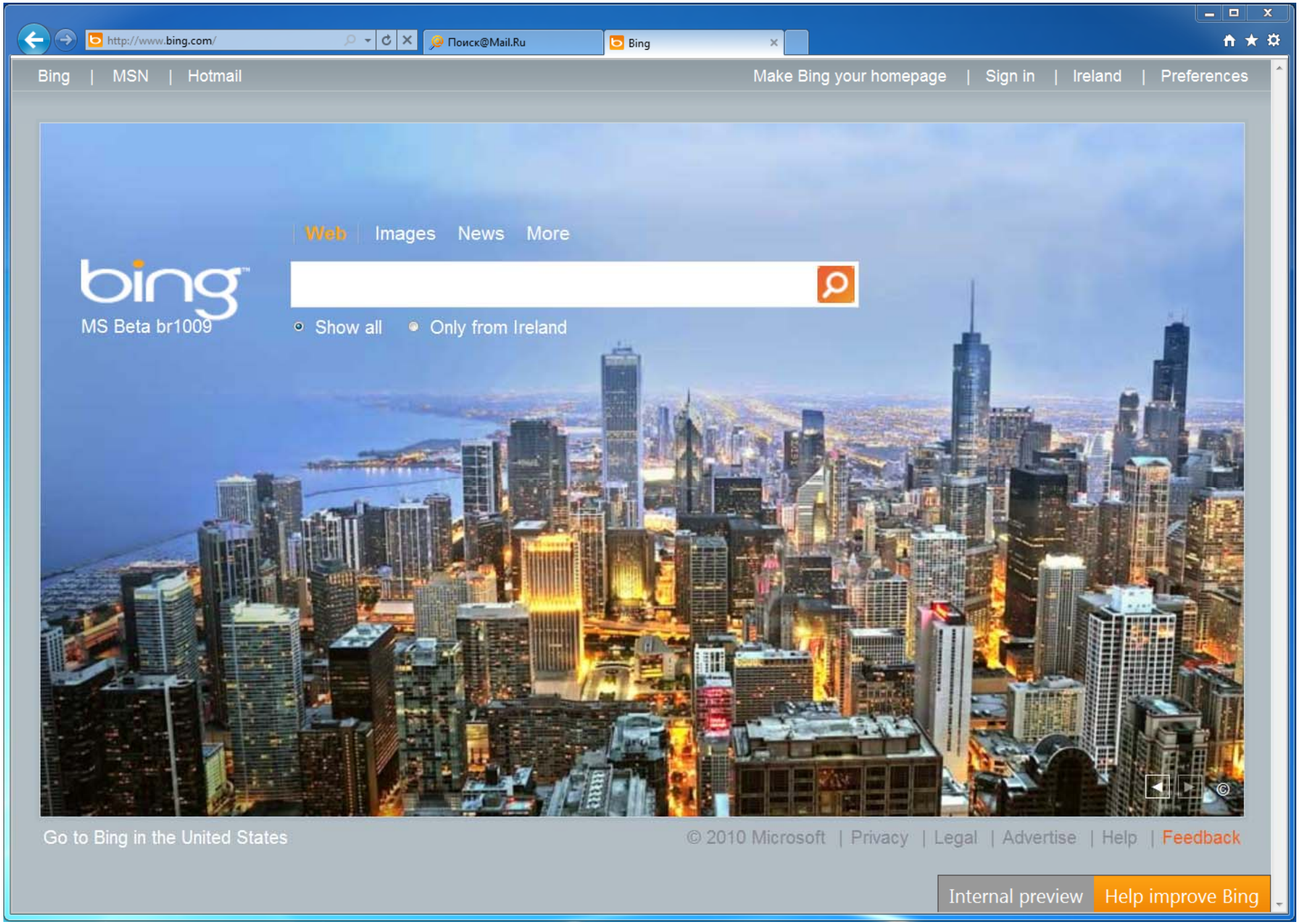
$$\mu\left(X^{\ell}\right) \qquad \sup_{f \in \mathbb{F}}$$

- Data-dependent analysis
  - Connectivity graph on algorithm set

Conclusion

- Can we use even more data mining stuff in Search Engine? Probably yes:
  - State-of-the-art BM25F ranking model
  - Neural Network for finest relevancy scores at top-N documents
  - Mix of linguistics with clustering for smart did-you-mean suggestions
  - Document preprocessing
- Should we be accurate introducing data mining models? Yes, **at least** because of overfitting.


http://go.mail.ru/
Поиск@Mail.Ru
Bing
На главную
Спутник@Mail.Ru
Тренды
Сообщество Поиска
Почта
Мой Мир
Войти
Поиск в Интернете
Картинки
Видео
Товары
Люди
Карты
Ответы
Работа
Словари
Hi-Tech
поиск@mail.ru
Any questions?☺
Найти
Золотые ворота
Коктебель, Чёртовы ворота Пушкин,
Сердоликовая бухта
© 2010 Mail.Ru
Помощь
Служба поддержки
Добавить сайт


http://www.bing.com/
Поиск@Mail.Ru
Bing
Bing | MSN | Hotmail
Make Bing your homepage | Sign in | Ireland | Preferences
Web Images News More
bing
MS Beta br1009
Show all
Only from Ireland
Go to Bing in the United States
© 2010 Microsoft | Privacy | Legal | Advertise | Help | Feedback
Internal preview
Help improve Bing

Conclusion: How it affects search?

- ToDo: find out how is used in Yahoo and Yandex.
  - Query categorization (what was user really searching for? Document? Information?)
  - User clustering? (results relevant for some users could be irrelevant for others)
  - Duplicate detection?
  - Information extraction from document, document markup
  - Page markup (in web-search)

Classification task examples

- Medical diagnostics
- [Классификация полезных искапаемых]
- Spam detection
- Off-line and on-line signature detection
- Speech recognition
- Credit scoring
- Churn prediction

Regression task examples

- Stock market and illegal behaviour detection
- Electricity consumption forecasting
- When to put money into MiniBANK?